

LOADING/STORING FUNCTION UNIT OF MICROPROCESSOR AND APPARATUS FOR INFORMATION PROCESSING

Publication number: JP7182167 (A)

Publication date: 1995-07-21

Inventor(s): UIRIAMU EMU JIYONSON; DEIBITSUDO BII UITSUTO;
MIYURARI CHINAKONDA +

Applicant(s): ADVANCED MICRO DEVICES INC +

Classification:

- international: G06F12/08; G06F9/312; G06F9/38; G06F12/08; G06F9/312;
G06F9/38; (IPC1-7): G06F12/08; G06F9/38

- European: G06F9/312; G06F9/38D; G06F9/38D4; G06F9/38E

Application number: JP19940260699 19941025

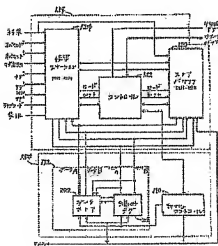
Priority number(s): US19930146376 19931029

Also published as:

EP0651323 (A1)
EP0651323 (B1)
US6296423 (B1)
US5878245 (A)
DE69433339 (T2)

Abstract of JP 7182167 (A)

PURPOSE: To perform plural load operations in parallel and to perform the store transfer operation by a super-scalar microprocessor provided with a load/store function unit and a corresponding data cache. CONSTITUTION: A load/store function unit 134 includes plural entries RS0 to RS3 of a holding station 124, which are accessed in parallel and are coupled to a data cache 150 in parallel, and a store buffer circuit 180 having plural buffer entries SB0 to SB3. Store buffer entries are constituted so as to provide such first-in first-out buffer that the output from a lower-order entry of the buffer is given as the input to a higher-order entry.



Data supplied from the espacenet database — Worldwide

(51) Int.Cl.⁶G 0 6 F 9/38
12/08

識別記号

3 7 0 A

庁内整理番号

C 7608-5B

FI

技術表示箇所

審査請求 未請求 請求項の数18 OL (全 26 頁)

(21) 出願番号 特開平6-260699

(22) 出願日 平成6年(1994)10月25日

(31) 優先権主張番号 1 4 6 3 7 6

(32) 優先日 1993年10月29日

(33) 優先権主張国 米国 (US)

(71) 出願人 591016172

アドバンスト・マイクロ・デバイス・
インコーポレイテッド
ADVANCED MICRO DEVI
CES INCORPORATED
アメリカ合衆国、94088-3453 カリフォ
ルニア州、サニイバベル、ビー・オー・ボ
ックス・3453、ワン・エイ・エム・ディ・
プレイス (番地なし)

(74) 代理人 弁理士 深見 久郎 (外3名)

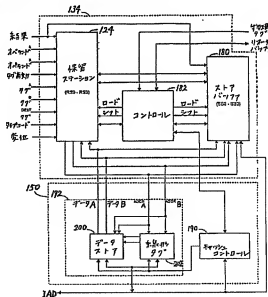
最終頁に続く

(54) 【発明の名称】 マイクロプロセッサのロード/ストア機能ユニットおよび情報処理用装置

(57) 【要約】

【目的】 ロード/ストア機能ユニットおよび対応するデータキャッシュを有するスーパースカラマイクロプロセッサによって、複数のロード動作を並列に行ない、かつストア転送動作を行なう。

【構成】 ロード/ストア機能ユニット(134)は並列にアクセスされデータキャッシュ(150)に並列に結合される保留ステーション(124)の複数個のエントリ(RS0-RS3)と、複数個のバッファエントリ(SB0-SB3)を有するストアバッファ回路(180)を含む。ストアバッファエントリはバッファの下位エントリからの出力が上位エントリへの入力として与えられる先入れ先出しバッファを与えるように構成される。



【特許請求の範囲】

【請求項1】 複数個のロード動作を並列に実行するためのロード機能ユニットであって、ロード動作を一時的に保持するための保留ステーション回路を含むロード機能ユニットを備え、前記保留ステーション回路は第1の保留ステーションエン트리と第2の保留ステーションエントリとを含み、前記第2の保留ステーションエントリは前記第1の保留ステーションエントリに結合され、かつ前記第1の保留ステーションエントリに保留ステーションエントリ出力を与え、さらにロード信号を並列に受取り、第1のロード信号を前記第1の保留ステーションエントリと前記第2の保留ステーションエントリとに与え、かつ第2のロード信号を前記第1の保留ステーションと前記第2の保留ステーションとに与える入力信号マルチプレクサ回路と、

前記第1の入力ロード信号と前記第2の入力ロード信号とのうちのどちらを前記第1の保留ステーションエントリおよび前記第2の保留ステーションエントリが取出すかを制御するためのロード制御回路とを含み、前記ロード機能ユニットはさらにストアアレイとデータキャッシュコントローラとを含むデータキャッシュを含み、前記ストアアレイは第1のデータキャッシュポートと第2のデータキャッシュポートとの各々を介して前記ロード機能ユニットの前記第1の保留ステーションエントリと前記第2の保留ステーションエントリとに結合され、前記ストアアレイは前記第1のロード信号と前記第2のロード信号とにตอบสนองしてデータを前記ロード機能ユニットに並列に与え、

前記データキャッシュコントローラは前記ロード制御回路に結合される、ロード機能ユニット。

【請求項2】 前記保留ステーション回路はデータキャッシュデータを並列に受取り、かつ前記データキャッシュデータを第1の結果バスと第2の結果バスとに並列に与える保留ステーションドライバ回路をさらに含む、請求項1に記載のロード機能ユニット。

【請求項3】 前記ロード制御回路はタイプコード一致信号にตอบสนองして前記第1の保留ステーションエントリと前記第2保留ステーションエントリのどちらによってどのロード信号が取出されるかを制御し、前記タイプコード一致信号は、タイプコードバスからのタイプコードが所定のロード機能ユニットタイプコードに一致する場合に前記ロード制御回路によって発生する、請求項1に記載のロード機能ユニット。

【請求項4】 前記保留ステーション回路は第3の保留ステーションエントリをさらに含み、前記第3の保留ステーションエントリは前記第2の保留ステーションエントリに結合されて前記第2の保留ステーションエントリに第3の保留ステーションエントリ出力を与え、前記第3の保留ステーションエントリは前記第1の保留ステーションエントリに結合されて前記第1の保留ステーション

エントリに前記第3の保留ステーションエントリ出力を与え、前記第1の保留ステーションエントリと前記第2の保留ステーションエントリのうちの一方は前記ロード制御回路の制御下で前記第3の保留ステーションエントリ出力を取出す、請求項1に記載のロード機能ユニット。

【請求項5】 前記保留ステーション回路は第4の保留ステーションエントリをさらに含み、前記第4の保留ステーションエントリは前記第3の保留ステーションエントリに結合されて前記第3の保留ステーションエントリに第4の保留ステーションエントリ出力を与え、前記第4の保留ステーションエントリは前記第2の保留ステーションエントリに結合されて前記第2の保留ステーションエントリに前記第4の保留ステーションエントリ出力を与え、

前記第3の保留ステーションエントリと前記第2の保留ステーションエントリのうちの一方は前記ロード制御回路の制御下で前記第4の保留ステーションエントリ出力を取出す、請求項4に記載のロード機能ユニット。

【請求項6】 前記保留ステーション回路は、前記第1の保留ステーションエントリと前記第2の保留ステーションエントリとにそれぞれ結合される第1の加算器回路と第2の加算器回路とをさらに含み、

前記第1の加算器回路および前記第2の加算器回路は前記ロード信号を受取って前記ロード信号に基づいてキャッシュアドレス信号を与え、前記キャッシュアドレス信号は前記データキャッシュストアアレイ内の第1の位置および第2の位置の各々にアクセスする、請求項1に記載のロード機能ユニット。

【請求項7】 前記第1の加算器回路および前記第2の加算器回路の各々は複数個のアドレス成分信号を受取り、論理アドレス信号を与えるための論理アドレス加算器と、前記論理アドレス信号とセグメントベース信号とを受取り、線形アドレスを与えるための線形アドレス加算器とを含む、請求項6に記載のロード機能ユニット。

【請求項8】 前記アドレス成分信号はAオペランド加算器信号と、Bオペランド加算器信号と、変位加算器信号とを含む、請求項7に記載のロード機能ユニット。

【請求項9】 前記第1の加算器回路は、Aオペランド信号とゼロ信号とを受取り、前記ロードコントロールからのアドレスモード制御情報にตอบสนองしてこれらの値のうちの1つを前記Aオペランド加算器信号として与えるためのオペランドマルチプレクサ回路と、Bオペランド信号と誤置列アドレス1信号とを受取り、前記ロードコントロールからのアドレスモード制御情報にตอบสนองしてこれらの信号のうちの1つを前記Bオペランド加算器信号として与えるためのBオペランドマルチプレクサ回路と、変位信号と4信号と5信号とを受取り、前記ロードコントロールからのアドレスモード制御情報

に広答してこれらの値のうちの1つを前記変位加算器信号として与えるための変位マルチプレクサ回路とをさらに含む、請求項8に記載のロード機能ユニット。

【請求項10】 前記第2の加算器回路は、Aオペランド信号とゼロ信号とを受取り、前記ロードコントロールからのアドレスモード制御情報に広答してこれらの値のうちの1つを前記Aオペランド加算器信号として与えるためのAオペランドマルチプレクサ回路と、Bオペランド信号と該並列アドレス1信号とを受取り、前記ロードコントロールからのアドレスモード制御情報に広答してこれらの信号のうちの1つを前記Bオペランド加算器信号として与えるためのBオペランドマルチプレクサ回路とをさらに含む、

変位信号は前記論理アドレス加算器に直接与えられる、請求項8に記載のロード機能ユニット。

【請求項11】 ストア転送動作を実行するためのストア機能ユニットであって、

ストア動作を保持するための第1および第2のストアバッファエントリ回路を含む、前記第2のストアバッファエントリは前記第1のストアバッファエントリに結合されて前記第1のストアバッファエントリに第2のストアバッファエントリ出力を与え、前記第1のストアバッファエントリは前記第2のストアバッファエントリに結合されて前記第2のストアバッファエントリに第1のストアバッファエントリ出力を与え、さらに前記第1のストアバッファエントリ出力を用いてストア転送動作を実行するために、前記第2のストアバッファエントリ回路が前記第1のストアバッファエントリ出力を取出すかどうかを制御するためのストアコントロールを含む、前記ストアコントロールは前記第1のストアバッファエントリ回路と前記第2のストアバッファエントリ回路とに結合される、ユニット。

【請求項12】 第3のストアバッファエントリ回路をさらに含む、前記第3のストアバッファエントリ回路は前記第2のストアバッファエントリ回路に結合されて前記第2のストアバッファエントリに第3のストアバッファエントリ出力を与え、前記第1のストアバッファエントリ回路は前記第3のストアバッファエントリ回路に結合されて前記第3のストアバッファエントリに第1のストアバッファエントリ出力を与え、かつ前記第2のストアバッファエントリ回路は前記第3のストアバッファエントリ回路に結合されて前記第3のストアバッファエントリに第2のストアバッファエントリ出力を与え、さらに前記ストアコントロールは前記第3のストアバッファエントリ回路に結合され、前記第1および第2のストアバッファエントリ出力を用いてストア転送動作を実行するために、前記第3のストアバッファエントリ回路が前記第1のストアバッファエントリ出力を取出すかまたは前記第2のストアバッファエントリ出力を取出すかを制御する、請求項11に記載のストア機能ユニ

ット。

【請求項13】 第4のストアバッファエントリ回路をさらに含む、前記第4のストアバッファエントリ回路は前記第3のストアバッファエントリ回路に結合されて前記第3のストアバッファエントリに第4のストアバッファエントリ出力を与え、前記第1のストアバッファエントリ回路は前記第4のストアバッファエントリ回路に結合されて前記第4のストアバッファエントリに第1のストアバッファエントリ出力を与え、前記第2のストアバッファエントリ回路は前記第4のストアバッファエントリ回路に結合されて前記第4のストアバッファエントリに第2のストアバッファエントリ出力を与え、かつ前記第3のストアバッファエントリ回路は前記第4のストアバッファエントリ回路に結合されて前記第4のストアバッファエントリに第3のストアバッファエントリ出力を与え、さらに前記第4のストアバッファエントリ回路に結合され、前記第1または前記第2のストアバッファエントリ出力を用いてストア転送動作を実行するために、前記第4のストアバッファエントリ回路が前記第1のストアバッファエントリ出力を取出すかまたは前記第2のストアバッファエントリ出力を取出すかを制御する、請求項12に記載のストア機能ユニット。

【請求項14】 前記第1のストアバッファエントリ回路および前記第2のストアバッファエントリ回路の各々はストアバッファエントリを保持するためのストアバッファレジスタ回路と、

どの信号が前記ストアバッファレジスタ回路に与えられて保持されるかを制御するためのストアバッファマルチプレクサ回路とを含む、請求項11に記載のストア機能ユニット。

【請求項15】 前記ストアバッファレジスタ回路は前記ストアバッファエントリのストアバッファデータエントリを保持するためのストアバッファエントリデータレジスタと、

前記ストアバッファエントリのストアバッファアドレスエントリを保持するためのストアバッファエントリアドレスレジスタと、

前記ストアバッファエントリのストアバッファタグエントリを保持するためのストアバッファエントリタグ部分とを含む、請求項14に記載のストア機能ユニット。

【請求項16】 前記ストアバッファマルチプレクサ回路は複数個のデータ信号を受取り、前記複数個のデータ信号のうちの1つを前記ストアコントロールの制御下で前記ストアバッファデータエントリとして与えるためのデータバイトマルチプレクサ回路と、

複数個のアドレス信号を受取り、前記複数個のアドレス信号のうちの1つを前記ストアコントロールの制御下で前記ストアバッファアドレスエントリとして与えるためのアドレスバイトマルチプレクサ回路と、

複数個のタグ信号を受取り、前記複数個のタグ信号のうちの少なくとも1つを前記ストアコントロールの制御下で前記ストアバッファタグエントリとして与えるためのタグマルチプレクサ回路とを含む、請求項15に記載のストア機能ユニット。

【請求項17】 キャッシュに対してロード動作およびストア動作を並列に実行する、マイクロプロセッサのロード/ストア機能ユニットであって、

ロード動作およびストア動作を一時的に保持するための保留ステーション回路を含み、前記保留ステーション回路は第1の保留ステーションエントリと第2の保留ステーションエントリとを含み、前記第1の保留ステーションエントリと前記第2の保留ステーションエントリとはデータキャッシュの第1のポートと第2のポートとに結合され、さらにストア動作を一時的に保持するためのストアバッファ回路を含み、前記ストアバッファ回路はストア動作を一時的に保持するための第1のストアバッファエントリと第2のストアバッファエントリとを含み、前記ストアバッファエントリのうちの少なくとも1つは前記保留ステーションエントリのうちの少なくとも1つに結合され、さらに前記保留ステーションエントリおよび前記ストアバッファエントリを制御するための制御回路を含み、前記制御回路は前記保留ステーション回路と前記ストアバッファ回路と前記データキャッシュとに結合される、ユニット。

【請求項18】 情報処理装置であって、情報を保持するための外部メモリと、プロセッサバスを介して主メモリに結合されるプロセッサとを含む、

前記プロセッサは前記外部メモリに結合されて前記情報を一時的に保持するためのキャッシュと、ロード動作およびストア動作を実行するためのロード/ストア機能ユニットとを含み、前記ロード/ストア機能ユニットはロード動作およびストア動作を一時的に保持するため保留ステーション回路を含み、前記保留ステーション回路は、前記データキャッシュの第1のポートと第2のポートとに結合された第1の保留ステーションエントリと第2の保留ステーションエントリとを含み、さらに第1のストアバッファエントリと第2のストアバッファエントリとを含んでストア動作を一時的に保持するためのストアバッファ回路を含み、前記ストアバッファエントリのうちの少なくとも1つは前記保留ステーションエントリのうちの少なくとも1つに結合され、さらに前記保留ステーション回路と前記ストアバッファ回路と前記データキャッシュとに結合されて、前記保留ステーションエントリおよび前記ストアバッファエントリを制御するための制御回路を含む、情報処理装置。

【発明の詳細な説明】

【0001】

【発明の背景】本発明はマイクロプロセッサに関し、よ

り特定的には、高性能なデータキャッシュおよびロード/ストア機能ユニットを有するマイクロプロセッサに関する。

【0002】マイクロプロセッサは、1つまたは非常に少数の半導体チップ上で実現されるプロセッサである。半導体チップ技術によりマイクロプロセッサ内の回路密度および速度が増大してきているが、マイクロプロセッサと外部メモリとの相互接続はパッケージング技術の制約を受けている。オンチップ相互接続にかかる費用は非常に安い、オフチップ接続の費用は非常に高い。マイクロプロセッサの性能を向上しようとするどのような技術も、パッケージング技術、およびプロセッサとその外部メモリとの物理的分離という制約内で、増大する回路密度および速度を利用しなければならない。回路密度が増大しているために、増々、設計が複雑になっているが、マイクロプロセッサの動作はユーザがマイクロプロセッサの使用法を理解できるように単純かつ明快なままでなくてはならない。

【0003】既存のマイクロプロセッサはスカラ計算に向けられているものが大半であるが、マイクロプロセッサの進化の点から言えば、スーパースカラマイクロプロセッサが論理的には次のステップである。スーパースカラという用語は、スカラ命令を同時に実行することによって性能を向上させるコンピュータ実現のことである。スカラ命令とは、典型的には汎用マイクロプロセッサで見られるタイプの命令である。今日の半導体処理技術を用いれば、かつては大規模な科学的プロセッサにしか適用できなかった高性能な技術を1つのプロセッサチップに組み入れることができる。しかしながら、大規模プロセッサに用いられる技術の多くは、スカラ計算には適さないか、またはマイクロプロセッサに用いるには高価すぎる。

【0004】マイクロプロセッサはアプリケーションプログラムを実行する。アプリケーションプログラムは命令群からなる。アプリケーションプログラムの実行においては、プロセッサはあるシーケンスで命令を取出しかつ実行する。たった1つの命令を実行するのに、命令を取出し、デコードし、そのオペランドをアセンブルし、命令によって特定される動作を実行し、かつ命令の結果を記憶に書込むといういくつかのステップがある。命令の実行は周期的クロック信号によって制御される。クロック信号の周期はプロセッササイクル時間である。

【0005】プロセッサがプログラムを完了するのにかかる時間は次の3つの要因、すなわちプログラムを実行するために必要な命令の数、命令を実行するために必要なプロセッササイクルの平均数、およびプロセッササイクル時間によって決定される。プロセッサの性能は所要時間を減じることによって向上するが、これには上の要因のうちの1つ以上のものが減らされる必要がある。

【0006】マイクロプロセッサの性能を向上させる一

方法としては、パイプライン処理と呼ばれる技術を用いて異なる命令のステップを重ねることである。命令をパイプライン処理するためには、パイプライン段と呼ばれる独立したユニットによって様々なステップの命令の実行が行われる。パイプライン段はクロック決めたレジスタによって分離される。異なる命令のステップが、異なるパイプライン段中で独立して実行される。パイプライン処理は、プロセッサが1度に1つ以上の命令を取扱うことを可能にすることによって、命令の実行に必要な合計時間は低減できないが、命令の実行に必要な平均サイクル数を低減する。このことはプロセッサのサイクル時間を目に見るほどは増大させることなく行われる。典型的には、パイプライン処理は1命令当たりの平均サイクル数を3分の1ほどにも低減する。しかしながら、分岐命令を実行する場合は、パイプラインは分岐動作結果がわかり正しい命令が実行用に取出されるまで停止することがあるかもしれない。これは分岐遅延ペナルティとして知られている。また、パイプライン段の数を増大すると、典型的には1命令当たりの平均サイクル数に関する分岐遅延ペナルティを増大させる。

【0007】プロセッサの性能を向上させる他の方法としては、マイクロプロセッサが命令オペランドをアセンブルし、かつ命令の結果を書込む速度を増大することである。これらの機能はそれぞれロードおよびストアと呼ばれる。これら両方の機能はマイクロプロセッサのデータキャッシュの使用に依存する。

【0008】初期のマイクロプロセッサの開発の間には、命令の実行時間と比べて命令を取出すのにかかる時間が長かった。このことが複合命令セットコンピュータ(CISC)プロセッサ開発の動機となった。CISCプロセッサは、利用可能な技術を用いるとすれば、1命令当たりのサイクル数は命令を取出すのにかかるサイクルの数に向上してほぼ決定されるという知識に基づいていた。性能の向上のために、CISCアーキテクチャの2つの主要な目標は、所与のタスクに必要な命令の数を低減することと、これらの命令を密度高くエンコードすることであった。パイプライン処理を用いると、デコードおよび実行サイクルは相対的に長い命令フェッチと通常は重なるため、命令をデコードしかつ実行するのにかかる平均サイクル数を増やすことによつて上記のような目標を達成してもよかった。このような前提のもとに、CISCプロセッサはプロセッサ内部のデコードおよび実行時間を犠牲にして密度高くエンコードされた命令を発展させた。複数サイクルの命令によって命令全体の数が減り、命令取出し時間が減るため全体の実行時間が短くなった。

【0009】1970年代後半から1980年代初めにかけて、メモ技術およびバッチング技術が急速に変化した。メモリ密度および速度は、キャッシュと呼ばれる高速ローカルメモリをプロセッサの近くに実現し得る

ほどまで増大した。キャッシュは、プロセッサが一時的に命令およびデータを記憶するために用いる。キャッシュを用いて命令がより迅速に取出される場合、以前は命令取出し時間内に隠れていたデコードおよび実行時間によって性能が限定される。命令の数は1つの命令を実行するのにかかる平均サイクル数ほどには性能に影響を及ぼさない。

【0010】命令の取出しが命令の実行よりそれほど時間がからならないようになるまでメモリおよびバッチング技術が向上したことは、縮小命令セットコンピュータ(RISC)プロセッサの開発の動機となった。性能の向上のために、RISCアーキテクチャの主たる目標は、命令の総数はいくぶん増やしても、命令の実行にかかるサイクルの数を減らすことである。1命令当たりのサイクル数と命令の数との間のトレードオフは1対1ではない。CISCプロセッサと比較して、RISCプロセッサは典型的には命令の数を30%から50%増やす一方で、3分の1〜5分の1にまで1命令当たりのサイクル数を減らす。RISCプロセッサは、コンパイラが全体の命令カウントを減らすことを助けるため、または1命令当たりのサイクル数を減らすことを助けるために、多数の汎用レジスタならびに命令およびデータキャッシュなどの補助的特徴に依存する。

【0011】典型的なRISCプロセッサはプロセッササイクル毎に1つの命令を実行する。スーパースカラプロセッサは、異なるパイプライン中で複数の命令を同時に実行できることに加え、同じパイプライン段中でも複数の命令を同時に実行できるため、パイプライン処理されたスカラーRISCプロセッサで可能であった以上に1命令当たりの平均サイクル数を低減する。スーパースカラという用語は、科学計算で一般的であるベクトルまたは配列に対する複数の同時演算とは異なり、スカラー量に対する複数の同時演算を強調するものである。

【0012】スーパースカラプロセッサは概念的には単純であるが、性能を向上させるためにはプロセッサのパイプラインの幅を広げる以上のことが必要である。パイプラインの幅を広げると1サイクルにつき1つ以上の命令を実行することが可能となるが、いかなる命令シーケンスもこの能力を利用できるとは限らない。命令は互いに独立しておらず相関しており、このような相関関係はいくつかの命令が同じパイプライン段を占有することを阻害する。さらに、命令をデコードしかつ実行するプロセッサの機構は、同時に実行可能な命令を発見する能力において大きく異なることがある。

【0013】スーパースカラ技術は、主として命令セットおよび他のアーキテクチャの特徴から独立したプロセッサの組織に関する。したがって、スーパースカラ技術の魅力の1つは、既存のアーキテクチャとコードレベルで互換性があるプロセッサを開発する可能性があることである。スーパースカラ技術の多くはRISCアーキテ

クチャまたはCISCアーキテクチャのどちらにも同じように十分に適用可能である。しかしながら、数多くのRISCアーキテクチャには規則性があるため、スーパーカラ技術はまずRISCプロセッサ設計に適用されている。

【0014】単一サイクルのデコードに適合しているというRISCプロセッサの命令セットの属性はまた、同じクックサイクル中に複数のRISC命令をデコードする場合にも適合するであろう。これらの属性には、一般的な3オペランドのロード/ストアアーキテクチャ、わずかに数命令分の長さしか持たない命令、わずかに数アドレスモードしか用いない命令、固定幅レジスタ、および命令フォーマット内のわずかに数箇所のレジスタ識別子に対する演算をする命令が含まれる。スーパースカラRISCプロセッサを設計する技術はニュージャージー州エングルウッド・クリフス (Englewood Cliffs, New Jersey) のプレントリス・ホール社 (Prentice-Hall, Inc.) (サイモン・アンド・シュスター (Simon & Schuster) の一部門ウィリアム・マイケル・ジョンソン (William Michael Johnson) による1991年の「スーパースカラマイクロプロセッサ設計 (Superscalar Microprocessor Design)」に記載されている。

【0015】RISCアーキテクチャとは対照的に、CISCアーキテクチャは多数の異なる命令フォーマットを用いる。幅広く使用されているCISCマイクロプロセッサアーキテクチャの1つはX86アーキテクチャである。このようなアーキテクチャは1386™マイクロプロセッサに最初に導入されたが、1486™マイクロプロセッサおよびペンティアム (Pentium™) マイクロプロセッサの両方の基礎となるアーキテクチャであり、これらのマイクロプロセッサはすべてカリフォルニア州サンタクララ (Santa Clara, California) のインテル (Intel) 社から入手可能である。X86アーキテクチャは、論理アドレス、線形アドレス、および物理アドレスの3つの異なるタイプのアドレスを提供する。

【0016】論理アドレスはセグメントベースアドレスからのオフセットを示す。実効アドレスと呼ばれるこのオフセットは、マイクロプロセッサが用いているアドレスモードのタイプに基づいている。これらのアドレスモードは、変位、ベース、インデックスおよびスケールの4つのアドレス要素の異なる組み合わせを与える。セグメントベースアドレスはセグメントを介してアクセスされる。より特定的には、セグメントレジスタ中に記憶されるセグメントはグローバルディスクリプタテーブル (GDT) 中の位置を示すインデックスである。GDT位置は、セグメントベースアドレスに対応する線形アドレスを記憶する。

【0017】論理アドレスと線形アドレス間の変換は、マイクロプロセッサが実モードであるのかまたは保護モードであるのかに依存する。マイクロプロセッサが実

モードの場合、セグメンテーションユニットはセグメントを4ビット左へシフトし、その結果をオフセットに加えて線形アドレスを形成する。マイクロプロセッサが保護モードの場合、セグメントユニットはセグメントが示した線形ベースアドレスをオフセットに加えて線形アドレスを与える。

【0018】物理アドレスはマイクロプロセッサのアドレスバス上に現われるアドレスであり、外部メモリを物理的にアドレス指定するために用いられる。物理アドレスは必ずしも線形アドレスには対応しない。ページングがイネーブルされなければ、32ビット線形アドレスは物理アドレスに対応する。もしページングがイネーブルされれば、線形アドレスは物理アドレスに変換されなくてはならない。この変換はページングユニットが実行する。

【0019】ページングユニットは線形アドレスを物理アドレスに変換するために2レベルの表を用いる。第1レベルの表はページディレクトリであり、第2レベルの表はページテーブルである。ページディレクトリは複数のページディレクトリエントリを含み、その各々のエントリはページテーブルのアドレスおよびページテーブルについての情報を含む。線形アドレスの上位10ビット (A22-A31) はページディレクトリエントリを選択するためのインデックスとして用いられる。ページテーブルは複数のページテーブルエントリを含み、各ページテーブルエントリはページフレームの実ページ番号と呼ばれるページフレームの開始アドレスと、ページについての統計的情報を含む。線形アドレスのアドレスビットA12-A21はページテーブルエントリの1つを選択するためのインデックスとして用いられる。ページフレームの開始アドレスは線形アドレスの低位2ビットと連結されて物理アドレスを構成する。

【0020】各メモリ動作毎に2レベルの表とアクセスすることはマイクロプロセッサの性能にかなり影響を及ぼすため、x86アーキテクチャは最も最近アクセスされたページテーブルエントリのキャッシュを設け、このキャッシュはトランザクションロックアップサイドバッファ (TLB) と呼ばれる。マイクロプロセッサはエントリがTLBにない場合だけページングユニットを使用する。

【0021】キャッシュを含むx86アーキテクチャに従う最初のプロセッサは486プロセッサであり、これは8KBバイト統合キャッシュを1つ含んでいた。ペンティアムプロセッサは分離した8KBバイト命令およびデータキャッシュを含む。486プロセッサキャッシュおよびペンティアムプロセッサキャッシュは物理アドレスを介してアクセスされるが、これらのプロセッサの機能ユニットは論理アドレスを用いて動作する。したがって、機能ユニットがキャッシュへアクセスする必要がある場合、論理アドレスは線形アドレスへ、その後物理アドレ

スへ変換されなくてはならない。

【0022】

【発明の概要】パラレルにアクセスされる複数個の保留ステーションエントリを含むロード部を有するマイクロプロセッサを提供することによって、複数個のロード動作をパラレルに実行することが可能であることがわかった。

【0023】バッファの下位エントリからの出力がバッファの上位エントリへの入力として与えられる。先入れ先出しバッファとして構成される複数個のストアバッファエントリを含むストア部を有するマイクロプロセッサを提供することによって、ストア転送動作を実行することが可能であることもわかった。

【0024】

【実施例】以下に、本発明を実行するための、考えられる最良のモードを詳細に説明する。以下の説明は本発明を例示するものであると意図し、限定的であるとは考えらるべきではない。

【0025】図1を参照して、本発明は、X86命令セットを実行するスーパースカラX86マイクロプロセッサ100の文脈において最良に理解することができる。マイクロプロセッサ100は486XLバスまたは他の従来のマイクロプロセッサバスを介して、物理的にアドレス指定される外部メモリ101に結合される。マイクロプロセッサ100はバイトキュー106に結合される命令キャッシュ104を含み、バイトキュー106は命令デコーダ108に結合される。命令デコーダ108はRISCコア110に結合される。RISCコア110は、シフトユニット130(SHF)、算術論理ユニット131、132(ALU0およびALU1)、特別レジスタブロック133(SRB)、ロード/ストアユニット134(LSSEC)、分岐セクション135(BRNSEC)、および浮動小数点ユニット136(FPU)などの多様な機能ユニットとともに、レジスタファイル112およびリオーダーバッファ114を含む。

【0026】RISCコア110は、命令デコーダ108とロード/ストアユニット134とに間に結合される変位および命令、ロードストア(INLS)バス119とともに、機能ユニットに結合されるAおよびBオペランドバス116と、タイプおよびディスパッチ(TAD)バス118と、結果バス140とを含む。AおよびBオペランドバス116はまた、レジスタファイル112とリオーダーバッファ114とに結合される。TADバス118はまた命令デコーダ108に結合される。結果バス140はまたリオーダーバッファ114に結合される。さらに、分岐セクション135は、リオーダーバッファ114と、命令デコーダ108と、命令キャッシュ104とにXターゲットバス103を介して結合される。AおよびBオペランドバス116は、4つのパラレル2ビット幅Aタグバスと、4つのパラレル2ビット幅

Bタグバスと、12ビット幅Aタグ有効バスと、12ビット幅Bタグ有効バスと、4つの4ビット幅先行タグバスと、4つの8ビット幅オペコードバスとともに、4つのパラレル4ビット幅Aオペランドバスと、4つのパラレル4ビット幅Bオペランドバスとを含む。タイプおよびディスパッチバス118は、4つの8ビット幅タイプコードバスと、1つの4ビット幅ディスパッチバスとを含む。変位およびINLSバス119は、2つの3ビット幅変位バスと、2つの8ビット幅INLSバスとを含む。

【0027】命令キャッシュ104に加えて、マイクロプロセッサ100はまたデータキャッシュ150(DC CACHE)と物理タグ回路162とを含む。データキャッシュ150はRISCコアのロード/ストア機能ユニット134と、プロセス内アドレスおよびデータ(IAD)バス102とに結合される。命令キャッシュ104はまたIADバス102に結合される。物理タグ回路162は、IADバスを介して命令キャッシュ104とデータキャッシュ150との両方と相互動作する。命令キャッシュ104およびデータキャッシュ150は両方とも線形にアドレス指定可能なキャッシュである。命令キャッシュ104およびデータキャッシュ150は物理的に離れているが、キャッシュは両方とも同じアーキテクチャを用いて構成される。

【0028】マイクロプロセッサ100はまた、メモリ管理ユニット(MMU)164とバスインタフェースユニット160(BIU)とを含む。TLB164は、IADバスと物理変換回路162とに結合される。バスインタフェースユニット160は、486XLバス等の外部マイクロプロセッサバスと共に、物理変換回路162とデータキャッシュ150とIADバス102とに結合される。

【0029】マイクロプロセッサ100は命令のシーケンスを含むコンピュータプログラムを実行する。典型的には、コンピュータプログラムはハードディスク、フロッピーディスクまたはコンピュータシステム中に位置する他の不揮発性記憶媒体に記憶される。プログラムが実行される場合、プログラムは記憶媒体から主メモリ101へロードされる。プログラムの命令および関連したデータが一旦主メモリ101に入ると、個々の命令の実行準備が行なわれ、最終的にはマイクロプロセッサ100によって実行される。

【0030】主メモリ101に記憶された後、命令はバスインタフェースユニット160を介して命令キャッシュ104へ伝えられ、そこで命令は一時的に保持される。命令デコーダ108は命令キャッシュ104から命令を取り出し、その命令を検査し、適切なアクションを決定する。たとえば、デコーダ108はある特定の命令がPOP, LOAD, STORE, AND, OR, EXOR, ADD, SUB, NOP, JUMP, 条件JUM

P (BRANCH)、その他の命令のいずれであるかを決定し得る。どの特定の命令が存在するとデコード108が決定するかに依存して、命令はRISCコア110の適切な機能ユニットにディスパッチされる。LOADおよびSTOREはロードストアセクション134にディスパッチされる主要な2つの命令である。ロード/ストア機能ユニット134には実行される他の命令にはPUSHおよびPOPがある。

【0031】典型的には、これらの命令は、OP CODE, OPERAND A, OPERAND B, DESTINATIONというフォーマットの複数のフィールドを含む。たとえば、命令ADD A, B, Cは、レジスタAの内容をレジスタBの内容に加えてその結果をレジスタCに置くことを意味する。LOADおよびSTORE動作は多少異なるフォーマットを使用する。たとえば、命令LOAD A, B, Cは、アドレスから取出したデータを結果バス上に置くことを意味し、ここでA、BおよびCはAオペランドバス、Bオペランドバスおよび変位バス上に位置するアドレス成分を表わし、これらのアドレス成分は組合わさって論理アドレスを与え、この論理アドレスはセグメントベースと組合わさって線形アドレスを与え、そこからデータが取出される。また、たとえば、命令STORE A, B, Cは、アドレスによって示された位置にデータを記憶することを意味し、ここでAはAオペランドバス上に位置する記憶データであり、BおよびCはBオペランドバスおよび変位バス上に位置するアドレス成分を示し、これらのアドレス成分は組合わさって論理アドレスを構成し、論理アドレスはセグメントベースと組合わさって線形アドレスを与え、そこからデータが記憶される。

【0032】OP CODEは、オペコードバスを介して命令デコーダ108からRISCコア110の機能ユニットへ与えられる。特定の命令のOP CODEを適切な機能ユニットに与えなくてはならないことに加えて、命令用に指定されたOPERANDを取出して機能ユニットに送らなくてはならない。もし特定のオペランドの値がまだ計算されていないれば、機能ユニットが命令を実行する前にその値を計算して機能ユニットに与えなくてはならない。たとえば、もし現在の命令が前の命令に依存しているとすれば、現在の命令が実行される前に前の命令の結果を求めなくてはならない。この状況は従属性と呼ばれる。

【0033】機能ユニットが特定の命令を実行するのに必要なオペランドは、レジスタファイル112またはリオーダーバッファ114のいずれかによってオペランドバスに与えられる。オペランドバスはオペランドを適切な機能ユニットへ送る。機能ユニットがOP CODE, OPERAND A, およびOPERAND Bを一旦受取ると、機能ユニットは命令を実行し、かつその結果を結果バス140上に置く。結果バス140はすべての

機能ユニットの出力およびリオーダーバッファ114に結合される。

【0034】リオーダーバッファ114は先入先出し(FIFO)装置として管理される。命令デコーダ108が命令をデコードすると、対応するエントリがリオーダーバッファ114中に割り当てられる。その後、命令の実行が完了すると、命令によって計算された結果値が割り当てられたエントリに書込まれる。もし命令に関連する例外がなく、かつ命令に影響を及ぼす可能性のある解決されていない分岐がなければ、この結果値はその後レジスタファイル112に書込まれ、命令は廃棄される。関連したエントリがリオーダーバッファ114の先頭に到達したときに命令が完了していかねば、命令が完了するまでリオーダーバッファ114の進行は停止される。しかしながら、追加のエントリの割り当ては継続される。

【0035】各機能ユニットは、まだ完了していない命令からのOP CODEを記憶するためにそれぞれ保留ステーション回路(RS)120-126を含むが、これはなぜならその命令用のオペランドは機能ユニットにはまだ入手できないからである。各保留ステーション回路は、後で保留ステーション回路に到着する欠けているオペランドの場所をとっておくとともに、命令のOP CODEを記憶する。この技術は、保留中の命令が保留ステーションでそのオペランドと組合わされている間に、マイクロプロセッサ100が他の命令の実行を継続することを可能にすることによって、性能を向上させる。

【0036】マイクロプロセッサ100は、デコーダ108をRISCコア110の機能ユニットから分離することによって順序外の発行ができる。より特定的には、リオーダーバッファ114および機能ユニットの保留ステーションは、効果的に分配命令ウィンドウを確立する。したがって、デコーダ108は、命令が即座に実行できないとしても命令のデコードを継続できる。命令ウィンドウは命令のプールとして機能し、機能ユニットがさらに続けて命令を実行するときに、このプールから命令を引出す。このように、命令ウィンドウはマイクロプロセッサ100にバックアップ能力を与える。従属性がクリアされオペランドが利用可能になると、機能ユニットはウィンドウ中のより多くの命令を実行し、かつデコーダはさらに多くのデコードされていない命令でウィンドウを満たし続ける。

【0037】マイクロプロセッサ100は性能を高めるためにRISCコアの分岐セクション135を使用する。分岐が起こると次の命令は分岐の結果に依存するため、プログラムの命令ストリーム中の分岐はマイクロプロセッサが命令を取出す能力を阻害する。分岐セクション135は、命令の取出の間に発生する分岐の結果を予測する。つまり、分岐セクション135は分岐が起こるかどうかを予測する。たとえば、前の分岐の結果の発行

履歴を維持するために、分岐ターゲットバッファが用いられる。この履歴に基づいて、取出されたある分岐の間に、取出された分岐命令がどの分岐をとるかの決定がなされる。もし例外または分岐の予測間違いがあれば、予測間違いの分岐命令に従って割当てられたリオーダバッファ114の内容は廃棄される。

【0038】図2を参照して、ロード/ストア機能ユニット134は、データキャッシュ150と相互動作し、かつすべてのLOA命令およびすべてのSTORE命令を実行する機能ユニットである。ロード/ストア機能ユニット134は、保留ステーション回路124と、ストアバッファ回路180と、ロードストアコントロール182とを含む。保留ステーション回路124は4つの保留ステーションエントリ(RS0-RS3)を含み、かつストアバッファ回路180は4つのストアバッファエントリ(SB0-SB3)を含む。

【0039】保留ステーション回路124はロード動作またはストア動作を実行するために必要なすべてのフィールドを保持する。データ要素はクロックサイクル毎に2つの保留ステーションエントリへ発行され、かつクロックサイクル毎に2つの保留ステーションエントリからリタイアできる。保留ステーション回路124は、4つの結果バスと、4つの41ビットAオペランドバスの40ビットと、4つの41ビットBオペランドバスの32ビットと、AおよびBタグ有効バスと、4つのAタグバスと、4つのBタグバスと、4つの行先タグバスと、4つのタイプコードバスと、2つの変位バスと、2つのINLSバスとともに、データキャッシュ150のポートAおよびBの32ビットデータ部分に結合される。保留ステーション回路124は40ビットAオペランドバスと、32ビット保留ステーションデータバス(RDATA A, RDATA Bのそれぞれ)と、12ビットAタグバス(TAG A)と、12ビットBタグバス(TAG B)とともに、2つの32ビットアドレスバス(ADDR A, ADDR B)を介してストアバッファ回路に結合され、2つのアドレスバスはまたデータキャッシュ150のポートAおよびBのアドレス部分に結合される。保留ステーション124は、保留ステーションロードバスおよび保留ステーションストアバスを介してコントロール182に結合される。

【0040】保留ステーション回路124に結合されることに加えて、ストアバッファ回路180は4つの結果バスに結合され、かつまたストアバッファロードバスおよびストアバッファストアバスを介してロードストアコントロール182に結合される。ストアバッファ回路180はまた16バス102に結合される。

【0041】保留ステーション回路124およびストアバッファ回路180に結合されることに加えて、ロードストアコントロール182は物理タグ回路162およびリオーダバッファ114に結合される。コントロール1

82はまたデータキャッシュ150のキャッシュコントロール190に結合される。

【0042】データキャッシュ150は線形にアドレス指定された4方向にインターリーブされた8Kバイト4方向セットアソシアティブキャッシュであり、これはクロックサイクル毎に2つのアクセスをサポートし、言い換えればデータキャッシュ150は二重の実行をサポートする。データキャッシュ150の各セットは128のエントリを含み、各エントリは16バイトブロックの情報を含む。各16バイトブロック情報は4つの個々にアドレス指定可能な32ビットバンクのラインに記憶される。データキャッシュ150に個々にアドレス指定可能なバンクを設けることによって、データキャッシュ150は2つのポートを設けることに関連したオーバーヘッドを必要とすることなく、2方向にアクセス可能なデータキャッシュとして機能する。データキャッシュ150はデータキャッシュポートAおよびデータキャッシュポートBを介して2方向にアクセス可能であり、こうしてデータキャッシュ150は2つのロード動作を同時に実行することができる。データキャッシュポートAは、データ部分DATA Aと、アドレス部分ADDR Aとを含む。データキャッシュポートBは、データ部分DATA Bと、アドレス部分ADDR Bとを含む。

【0043】データキャッシュ150はデータキャッシュコントロール190とデータキャッシュアレイ192とを含む。データキャッシュコントロール190はデータキャッシュ150の様々な動作を統制するための制御信号を与える。データキャッシュアレイ192はデータキャッシュコントロール190の制御下でデータを記憶する。データキャッシュアレイ192はデータ記憶アレイ200と線形タグおよび状態アレイ202との2つのアレイで構成される。データキャッシュアレイ200はDATA AとDATA Bとの2つのデータ信号をロード/ストア機能ユニット134に与える。線形タグアレイ202はロード/ストア機能ユニット134から与えられる2つの線形アドレスADDR AおよびADDR Bを受け、かつ2つの4ビットタグヒット信号COL HIT A-03およびCOL HIT B-03をキャッシュアレイ200に与える。線形アドレスADDR AおよびADDR Bはまたデータストアアレイ200に与えられる。

【0044】ロード動作の間、ロードストア機能ユニット134の保留ステーション回路124はアドレスをデータキャッシュ150に与える。もしこのアドレスがキャッシュヒットを発生すれば、データキャッシュ150はストアアレイ200の対応するバンクおよびブロック中に記憶されているデータを保留ステーション回路124に与える。もしアドレスがポートAを介してデータキャッシュ150に与えられると、データはポートAを介して保留ステーション回路124に与えられる。代替的

に、もしアドレスがポートBを介してデータキャッシュ150に与えられると、データはポートBを介して保留ステーション回路に与えられる。ポートAおよびポートBを介して同時にアドレスがデータキャッシュ150に与えられ、かつデータキャッシュ150からデータを受取る。

【0045】記憶動作の間、記憶データは保留ステーション回路124からストアバッファ回路180に与えられる。記憶動作が解除されると、記憶されているデータおよびそれに対応するアドレスは1ADバスを介してデータキャッシュ150に与えられる。

【0046】図3を参照して、保留ステーション回路124は先入れ先出し(FIFO)バッファとして機能するデュアルアクセス保留ステーションである。保留ステーション回路124は、入力0マルチプレクサ回路206と、入力1マルチプレクサ回路208と、4つの保留ステーションエン트리RS0 210、RS1 211、RS2 212およびRS3 213とともに、保留ステーション0加算器回路216と、保留ステーション1加算器回路218と保留ステーションドライバ回路200とを含む。

【0047】マルチプレクサ回路206および208は、入力として、4つのAオペランドバスと、4つのBオペランドバスと、AおよびBタグ有効バスと、4つのAタグバスと、4つのBタグバスと、4つの先行タグバスと、4つのオペコードバスと、2つのINLSバスと、2つの変位バスとを受ける。マルチプレクサ回路206および208はまた、ロード記憶コントローラ182からバス選択信号を受ける。バス選択信号はタイプコードの一致に基づいて発生する。

【0048】タイプコードの一致は、4つのタイプコードバスのうちの1つの上のタイプコードがロード記憶機能ユニットに割当てられたタイプコードに対応する場合に発生する。タイプコードが一致する場合、ロード記憶コントローラ182はどのバスから情報が取出されるべきかを示すバス選択信号を発生する。保留ステーション回路124は2つのバスから同時に信号を取出すことができる。したがって、入力0マルチプレクサ206および入力1マルチプレクサ208のそれぞれについてロード記憶コントローラ182によって第1の組のバス選択信号と、第2の組のバス選択信号とが発生する。

【0049】第1の組のバス選択信号の制御下で、マルチプレクサ回路206は、第1のマルチプレクスされた保留ステーション入力信号(INPUT 0)を与え、これは保留ステーションへの入力信号として与えられる。INPUT 0信号は、Aオペランドバスのうちの1つからの信号と、Bオペランドバスのうちの1つからの信号と、Aタグバスのうちの1つからのタグと、対応するタグ有効バスからのAタグに対応するタグ有効ビットと、Bタグバスのうちの1つからのタグと、対応する

タグ有効バスからのBタグに対応するタグ有効ビットと、先行タグバスのうちの1つからの先行タグと、オペコードバスのうちの1つからのオペコードと、変位バスのうちの1つからの変位とを含む。第2の組のバス選択信号の制御下で、マルチプレクサ回路208は、第2のマルチプレクスされた保留ステーション入力信号(INPUT 1)を与え、これは保留ステーションへの第2の入力信号として与えられる。INPUT 1信号は、Aオペランドバスのうちの1つからの信号と、Bオペランドバスのうちの1つからの信号と、Aタグバスのうちの1つからのタグと、対応するタグ有効バスからのAタグに対応するタグ有効ビットと、Bタグバスのうちの1つからのタグと、対応するタグ有効バスからのBタグに対応するタグ有効ビットと、先行タグバスのうちの1つからの先行タグと、オペコードバスのうちの1つからのオペコードと、変位バスのうちの1つからの変位とを含む。

【0050】保留ステーションエン트리210-213の各々は、ロードおよびシフトビットのそれぞれとともに、2つの入力信号INPUT 0およびINPUT 1を並列に受ける。保留ステーションエン트리210-213はまた、4つの結果バスの各々から入力を受け、これらの結果バス入力はエントリのAオペランド部分およびBオペランド部分のみに与えられる。AオペランドタグおよびBオペランドタグに基づいてこれらの結果バスから情報が取出される。たとえば、Aオペランドタグが先行タグバスのうちの1つの上にある情報に対するヒットを与えると、対応する結果バスからの情報が取出され、かつ保留ステーションエントリのAオペランドフィールド中へロードされる。

【0051】さらに、保留ステーションエントリRS0は保留ステーションRS1またはRS2のいずれから保留ステーションエントリを受け、保留ステーションエントリRS0はRS0保留ステーションエントリの一部(Aオペランド部分)をRDATA A信号としてストアバッファ回路180に与え、かつすべてのRS0保留ステーションエントリをR0加算器216に与える。R0加算器216はこの保留ステーションエントリを用いてADDR A信号を発生する。保留ステーションエントリRS1は保留ステーションRS2およびRS3から保留ステーションエントリを受け、保留ステーションエントリRS1はRS1保留ステーションエントリの一部(Aオペランド部分)をRDATA B信号としてストアバッファ回路180に与え、かつすべてのRS1保留ステーションエントリをR1加算器218に与える。R1加算器218はこの保留ステーションエントリを用いてADDR B信号を発生する。保留ステーションRS2は保留ステーションRS3から保留ステーションエントリを受け、保留ステーションエントリRS2はRS2保留ステーションエントリを保留ステーションRS1お

およびRS0に与える。保留ステーションRS3はRS3保留ステーションエントリを保留ステーションRS2およびRS1に与える。

【0052】パラレル転送構造とともに保留ステーションからの並列の入力および出力を与えることによって、保留ステーション回路124は1サイクル当たり1つまたは2つのロード動作を実行し得る。より特定的には、ロードおよびシフト信号を用いて、コントローラ182は、1つまたは2つの保留ステーションエントリが所与のサイクル中にロードまたはシフトされるように保留ステーションエントリのロードおよびシフトを制御する。

【0053】1サイクル当たり1つの保留ステーションエントリが実行されている場合、保留ステーションRS0はロード動作およびストア動作の両方のために保留ステーションエントリをRS0加算器回路216に与え、さらに、RS0はストア動作のために保留ステーションエントリをストアバッファ180に与える。保留ステーションRS1は保留ステーションエントリを保留ステーションRS0に与え、保留ステーションRS2は保留ステーションエントリを保留ステーションRS1に与え、かつ保留ステーションRS3は保留ステーションエントリを保留ステーションRS2に与える。ロード動作のためには、RS0加算器回路216によって発生したアドレスに対応するデータがドライバ回路220に与えられる。

【0054】1サイクル当たり2つの保留ステーションエントリが実行されている場合、保留ステーションRS0およびRS1はロード動作およびストア動作の両方のために各保留ステーションエントリを加算器回路216、218に与える。保留ステーションRS2およびRS3は保留ステーションエントリを保留ステーションRS0およびRS1のそれぞれに与える。ロード動作のためには、RS0およびRS1加算器回路によって発生したアドレスに対応するデータがデータキャッシュ150からDATA AおよびDATA Bとして与えられる。1サイクル当たり2つの保留ステーションエントリが実行され、かつ一方の動作はロードであり他方の動作はストアである場合、ストア動作がそこから実行される保留ステーションエントリがストアバッファ180に与えられる。

【0055】ロードによりキャッシュミスが発生すれば、ロードミisalゴリズムを実行しなくてはならない。このロード動作は仮想的なものであるため、ミス動作はロードが次にリタイアするROPとなるまで開始されない。このため、ロードはRS0保留ステーション中で留まり、リオーガバッファからの解除信号を待つ。ステータス表示は先行タグとともにこの状態を示すためにリオーガバッファにドライババックされる。

【0056】図4を参照して、各保留ステーションエン

トリ124は、保留ステーションエントリ有効ビット(v)と、40ビットAオペランドフィールドと、32ビットBオペランドフィールドと、32ビット変位フィールドと、4ビット先行タグ(DTAG)フィールドと、8ビットオペコードフィールドと、8ビット追加オペコード情報(INLS)フィールドとを含む。さらに、各保留ステーションエントリはまた、4ビットAオペランド上位バイトタグ(ATAGU)と、4ビットAオペランド中位バイトタグ(ATAGM)と、4ビットAオペランド下位バイトタグ(ATAGL)と、4ビットBオペランド上位バイトタグ(BTAGU)と、4ビットBオペランド中位バイトタグ(BTAGM)と、4ビットBオペランド下位バイトタグ(BTAGL)とを、対応するAオペランドタグ有効ビットおよびBオペランドタグ有効ビットとともに含む。各保留ステーションエントリはまた、対応する取消ビット(C)を含む。

【0057】Aオペランド上位、中位および下位バイトタグは、整数オペランドの上位、中位および下位部分のタグである。蓋数オペランドはここのように除算されるが、これはなぜならx86アーキテクチャでは、x86整数の下位半ワードの上位バイトもしくは下位バイト、下位半ワード、または32ビット倍長語全体のいずれかを参照することが可能なためである。したがって、MおよびLは下位半ワードの上位バイトおよび下位バイトを示し、かつUはBオペランド用の上位半ワードおよびAオペランド用の残りの上位ビット(Aオペランドの残りの部分は16ビットまたは24ビットのいずれかであり得るため)を示す。下位半ワードを参照する場合、タグおよびMタグは同じ値に設定される。保留ステーションエントリ中で保真中である32ビット値を参照する場合、3つのタグすべてが同じ値に設定される。

【0058】取消ビットは特定のオペコードが取消されることを示し、このビットは何らかのオペコードが予測誤り分岐内にある場合に設定される。オペコードが取消されるのは、実行されるストアがデータキャッシュ150にストアされるエントリの状態を更新するときに、データキャッシュ150中でヒットした取消されたストアがストアバッファ回路180に入ることを防止するためである。取消されたロードはデータキャッシュ150中にヒットがあっても結果を戻すだけであり、ロードはどの状態も更新しないために問題とはならない。

【0059】保留ステーションエントリの保留ステーションエントリ有効ビットは、INPUT 0入力信号およびINPUT 1入力信号のディスパッチ有効ビット部分に結合される。ディスパッチバスに結合される各入力信号有効ビットは、ディスパッチ有効ビットがセットされるときにセットされる。保留ステーションエントリのAオペランドフィールドはINPUT 0入力信号およびINPUT 1入力信号のAオペランド部分に結合される。保留ステーションエントリのBオペランドフィ

ールドはINPUT 0入力信号およびINPUT 1入力信号のBオペランド部分に結合される。保留ステーションエントリの変位フィールドはINPUT 0入力信号およびINPUT 1入力信号の変位部分に結合される。保留ステーションエントリの先行タグフィールドはINPUT 0入力信号およびINPUT 1入力信号の先行タグ部分に結合される。保留ステーションエントリのオペコードフィールドはINPUT 0入力信号およびINPUT 1入力信号のオペコード部分に結合される。保留ステーションエントリの追加オペコード情報(INLS)フィールドはINLSバスを介してINPUT 0入力信号およびINPUT 1入力信号のINLS部分に結合される。

【0060】保留ステーションエントリのAオペランド上位バイトタグと、中位バイトタグと、下位バイトタグとはINPUT 0入力信号およびINPUT 1入力信号のAタグ部分に結合される。Bオペランド上位バイトタグと、中位バイトタグと、下位バイトタグとはINPUT 0入力信号およびINPUT 1入力信号のBタグ部分に結合される。保留ステーションエントリのAオペランドタグ有効ビットおよびBオペランドタグ有効ビットはINPUT 0入力信号およびINPUT 1入力信号のタグ有効部分に結合される。保留ステーションエントリの取消ビットは、ロードストアコントロール182に結合され、かつリオーダーバッファ114および分岐セクション135から受取った制御情報に基づいてセットされる。

【0061】ロードストアコントロール182によって発生したタイプ一致信号は、何らかの命令がロードストア機能ユニットに送られたかどうかを決定する。より特定のには、ロードストアコントロール182がロードストア機能ユニットタイプコードが4つのTADバスのうちの1つに与えられたタイプコードと一致すると判断した場合、ロードストアコントロール182はINPUT 0信号のためにその特定のディスパッチ位置を選択する。ロードストアコントロール182がロードストア機能ユニットタイプコードが4つのTADバスのうちの別のものによって与えられたタイプコードと一致すると判断すれば、ロードストアコントロール182はINPUT 1信号のためにその特定のディスパッチ位置を選択する。

【0062】図5を参照して、RS0加算器回路216は保留ステーション210からアドレス成分を受け、かつ線形アドレス信号ADDR Aおよび有効セグメントアクセス信号を与える。RS0加算器回路216は、論理アドレス加算器240と線形アドレス加算器242とを含む。論理アドレス加算器240は論理アドレスを線形アドレス加算器242に与える。論理アドレス加算器240はAオペランドマルチプレクサ244からAオペランド加算器信号を受け、Bオペランドマルチプレクサ

246からBオペランド加算器信号を受け、かつ変位マルチプレクサ248から変位加算器信号を受ける。

【0063】オペランドマルチプレクサ回路244は量ゼロを受けるとともに、保留ステーションエントリ210からAオペランドを受け、マルチプレクサされAオペランド加算器信号として与えられる値は、ロードストアコントロール182から受取ったアドレスモード制御情報によって決定される。Bオペランドマルチプレクサ回路246はシフト回路247からスケールされたBオペランドを受ける。Bオペランドは、INLSバスを介して命令デコーダ108から受取ったスケール信号に基づいてスケールされる。Bオペランドマルチプレクサ回路246はまた、ロードストアコントロール182の制御下で開始アドレスレジスタ249にストアされる開始アドレスと、以前の誤置列したアクセスから誤置列したアクセスレジスタ451にストアされる誤置列したアクセス1アドレスとを受ける。マルチプレクサされBオペランド加算器信号として与えられる値はアドレスモード制御情報によって決定される。変位マルチプレクサ回路248は保留ステーションエントリ210から変位アドレス成分を受ける。変位マルチプレクサ回路248はまた、量4、5、-4、および-2を受ける。マルチプレクサされかつ変位加算器信号として与えられる値はアドレスモード制御情報によって決定される。

【0064】整列したアクセスロード動作については、Aオペランドがマルチプレクサ244によって選択され、Bオペランドがマルチプレクサ246によって選択され、かつ変位が248によって選択される。誤置列したアクセスロード動作、つまり倍長境界線を横切るアクセスについては、第1の誤置列したアクセスアドレスがノーマルロード動作として発生し、かつ加算器240は誤置列したアクセス1アドレスを発生する。誤置列したアクセス1レジスタ251はこの誤置列したアクセス1アドレスを保持する。次のクロックサイクルでは、Aオペランドマルチプレクサ244によって値0が選択され、Bオペランドマルチプレクサ246によって値4が選択され、かつマルチプレクサ248によって誤置列したアクセス1アドレスが選択され、こうして加算器240が誤置列したアクセス1アドレスに量4を加える。複数ROP動作、たとえば64ビットロード動作については、ノーマルロード動作として第1のアクセスアドレスが発生され、かつ加算器240は複数ROP開始アドレスを発生する。開始アドレスレジスタ249はこの開始アドレスを保持する。第2のROPがアクセスされる場合、マルチプレクサ248からの開始アドレスと、マルチプレクサ246からの値4とを加えることによって第2のROPアドレスが形成される。80ビット複数ROP動作については、マルチプレクサ246によって値5が与えられる。各複数ROP動作は誤置列する場合があり、この場合は、開始アドレスは誤置列したアクセス

ドレス1と同じ働きをする。PUSH動作については、動作のアクセスサイズに依存してBオペランドアドレスからある値が減らされる。もしアクセスサイズが倍長語であれば、値4が減算され、もしアクセスサイズが語であれば、値2が減算される。シフト回路27を制御するスケーリング係数がNLS情報に基づいてロードストアコントロール182によって発生する。

【0065】加算器回路216はまた、セグメントディスクリプタアドレス250と、リミットチェック回路252を含む。セグメントディスクリプタアドレス250はセグメントリミット信号をリミット回路252に与え、セグメントベースアドレス信号を加算器回路242に与える。リミットチェック回路252はまた、論理加算器240から論理アドレスを受け、かつ論理アドレスがセグメントディスクリプタアドレス250によって与えられるリミットで説明されるようなセグメントリミット内であることを示す有効セグメントアクセス信号を与える。

【0066】加算器回路240は、Aオペランド加算器信号とBオペランド加算器信号と変位加算器信号とを受け、これらの信号を加算して論理アドレス信号を与える。加算器回路242はセグメントディスクリプタアドレス250から受けたセグメントベースアドレスを論理アドレスに加えて線形アドレスを与える。

【0067】RS1加算器218はRS0加算器と同様であるが、ただしRS0保留ステーションを用いる場合にのみ整列していないアクセスが実行されるため、RS1加算器218はマルチプレクサ248を含まない。RS1加算器218中では、変位加算器信号として変位が直接加算器240に与えられる。さらに、未整列のアクセスは実行されないため、RS1加算器218用のマルチプレクサ246には値4および値5は与えられない。

【0068】図6を参照して、ストアバッファは、Aポートマージ回路306およびBポートマージ回路308とともに、4つのストアバッファエントリSB0-SB3、SB1-SB2、SB3、およびSB0-SB3を含む。ポートマージ回路306はデータキャッシュ150からAポートデータ信号を受け、かつ保留ステーション回路124の保留ステーションエントリRS0からAポートデータ信号を受け、さらにこれらの信号をマージしてストアバッファエントリSB0-SB3にマージされたAデータ信号を与える。Bポートマージ回路308はデータキャッシュ150からBポートデータ信号を受け、かつ保留ステーション回路124の保留ステーションエントリRS1からBポートデータ信号を受け、さらにこれらの信号をマージしてストアバッファエントリSB0-SB3にマージされたBデータ信号を与える。マージ回路306、308を設けることにより、ステアリング機能を与えられる。

【0069】たとえば、4バイトDATA A信号のうち1バイトが、保留ステーション回路124によって

与えられる場合に更新されているかもしれない。この更新されたバイトは、データキャッシュ150によって与えられたDATA A信号からの3つの残りのバイトとマージされる。マージ回路306、308は、アクセスサイズ、線形アドレスの最下位2ビット、およびアクセスが誤整列されたアクセス1であるかまたは誤整列されたアクセス2であるかに基づいて、ロードストアコントロール182によって制御される。読出修正送動作としてストアが行なわれるため、マージ回路306、308によって与えられるステアリング機能が可能である。このステアリング機能を与えることにより、データキャッシュ150はデータキャッシュ150へのすべてのアクセスが32ビット倍長語アクセスであるため複雑なステアリング回路を必要としない。さらに、ストアバッファエントリ中にある情報はすべてデータキャッシュ150にストアされる情報を反映しているため、ロードストア機能ユニット134がロード転送動作を行なうことが可能となる。ロード転送動作においては、ストアバッファエントリをアクセスすることによってストアが実際にデータキャッシュ150中にストアされる前にロードが実行されてもよく、ロード転送によりクイックロードのクリティカルなタイミング経路からストア動作を取除く。

【0070】各ストアバッファエントリはまた、4つの結果バスから入力信号を受け、保留ステーション124からADDR Aアドレス信号およびADDR Bアドレス信号を受け、かつ保留ステーション124からTAG Aタグ信号およびTAG Bタグ信号を受取ることも、ロードストアコントロール182から制御信号を受取る。これらの制御信号はロード信号およびシフト信号を含む。さらに、ストアバッファエントリSB0はストアバッファエントリSB1からの出力を受け、かつストア出力をIADバス102に与える。ストアバッファエントリSB1はストアバッファエントリSB2から出力されたストアバッファエントリを受け、かつまたストアバッファエントリSB0から出力されたストアバッファエントリを受け、さらにストアバッファエントリ出力をSB0に与える。ストアバッファエントリSB2は、ストアバッファエントリSB3から出力されたストアバッファエントリを受け、かつまたストアバッファエントリSB0およびSB1からエントリを受け、さらにストアバッファエントリ出力をSB1に与える。ストアバッファエントリSB3は、ストアバッファエントリSB0、SB1およびSB2から出力されたストアバッファエントリを受け、かつストアバッファエントリ出力をSB2に与える。

【0071】ストアバッファエントリSB1-SB3に下位ストアバッファエントリからのフィードバックを与えることにより、ストア転送動作が可能となる。たとえば、ストアバッファエントリSB0を上位ストアバッファ

ァエントリSB1-SB3に与えることによって、これらのストアバッファエントリが同じ線形アドレスを有する場合にSB0ストアバッファエントリと上位エントリとを組合わせることができる。さらに、ストアバッファエントリがストアされる場合はエントリに対するあらゆる修正を含む。ストア転送機能については以下により詳しく説明する。

【0072】ストア転送により、前のストアがデータキャッシュ150にストアされるまで保留ステーションを停止させることなくシステムが動作できる。x86アーキテクチャにおいては、かなりの数の連続したバイトのアクセスが発生するため、ストア動作に対するロード動作の従属性を排除くことによって、ロードが行なわれる速度がストア転送によって大幅に上昇する。

【0073】図7を参照して、ストアバッファ回路180の各ストアバッファエントリSB0-SB3はストアバッファエントリ339で説明した情報を含む。ストアバッファエントリ339は32ビットデータ倍長語340と、タグ部分341と、32ビット線形アドレス342と、制御情報部分344とを含む。データ倍長語340は4つのデータバイト、つまりデータバイト0-データバイト3を含む。

【0074】タグ部分341はデータバイト0-3に対応する4つのバイトタグ部分を含む。バイト0タグ部分はバイト0タグ(TAG BYTE 0)と、バイト0制御ビット(B0)と、バイト0タグ有効ビット(TV)とを含む。バイト1タグ部分は、バイト1タグ(TAG BYTE 1)と、バイト1制御ビット(B1)と、バイト1タグ有効ビット(TV)とを含む。バイト2タグ部分は、バイト2タグ(TAG BYTE 2)と、バイト2制御ビット(B0, B1)と、バイト2タグ有効ビット(TV)とを含む。バイト3タグ部分は、バイト3タグ(TAG BYTE 3)と、バイト3制御ビット(B0, B1)と、バイト3タグ有効ビット(TV)とを含む。

【0075】バイトタグTAG BYTE 0-3は結果バスからデータバイト0-3を取出すためのタグを与える。バイト制御ビットはどの結果バスバイトからデータバイトが取出されるべきかを示す。より特定には、バイト0制御ビットB0がセットされている場合は、データが結果バスバイト1から転送されるべきであり、もしバイト1制御ビットB1がクリアされていると、データが結果バスバイト1から転送されるべきであることを示す。バイト2制御ビットB1がセットされている場合は、結果バスバイト0からデータが転送されるべきであることを示し、バイト2制御ビットB0がセッ

トされている場合は、結果バスバイト0からデータが転送されるべきであることを示し、もしバイト2制御ビットB0およびB1がクリアされていると、データは結果バスバイト2から転送されるべきである。バイト3制御ビットB1がセットされる場合は結果バスバイト1からデータが転送されるべきであることを示し、かつバイト3制御ビットB0がセットされている場合は結果バスバイト0からデータが転送されるべきであることを示し、もしバイト3制御ビットB0およびB1がクリアされていると、データは結果バスバイト3から転送されるべきである。バイトタグ有効ビットTVは対応するタグフィールドが有効バイトタグを含むことを示す。

【0076】ストアバッファタグはメモリ中の実際のバイト位置を示すが、保留ステーションタグではタグと位置との間には1対1の対応は存在しない。保留ステーションタグを用い、タグおよびMタグはストアバッファタグ内の任意の場所にマッピングすることができる。有効タグのついた未整列のアクセスはストアバッファの中へは入れない。未整列のアクセスについては、保留ステーションエントリRS0およびRS1は保留ステーションが有効データを受取るまで待機し、その後データは2つのストアバッファエントリとしてストアバッファに与えられる。

【0077】制御部分344はストアバッファエントリ有効ビット(V)と、2ビット未整列アクセス制御信号(UA)と、書込保護ビット(WB)と、キャッシュ不可ストアビット(NC)と、入力/出力アクセスビット(IO)と、浮動小数点更新ポインタビット(PP)と、物理アクセスビット(P)と、ロックされたアクセスビット(L)と、2ビットカラム表示ビット(C1)とを含む。ストアバッファエントリ有効ビットは、特定のエントリが有効である。つまりこのストアバッファエントリ中に何らかの有効な情報がストアされていることを示す。未整列のアクセス制御信号は、未整列のアクセスのどの部分、つまり第1の部分または第2の部分のどちらがエントリにストアされるかを示す。キャッシュ不可ストアビットは、ストアエントリがキャッシュ不可なためエントリをデータキャッシュ150に書込むことができないことを示す。I/Oアクセスビットは外部インタフェースに対してI/Oアクセスが発生していることを示す。物理アクセスビットは、ストアアドレスが物理アドレスであるため、メモリ管理ユニットが線形-物理変換をバイパスすべきであることを示し、これはロードストア機能ユニットがメモリ管理ユニット164のページディレクトリまたはTLBのいずれかを更新している場合に発生する。ロックされたアクセスビットは、以前のロードによってロックされているかもしれない外部バスのロックを外すことを示す。カラム表示信号は、データキャッシュの4つのカラムのうちの書込まれた1つを示し、このためストア動作を実行する際にデー

キャッシュ150中でカラムルックアップを実行する必要があることを示す。

【0078】図8を参照して、ストアバッファエントリ回路SB2 302が各ストアバッファエントリ回路の一例として示される。ストアバッファエントリ回路302は、ストアバッファエントリ339のデータバイト0-3に対応するストアバッファエントリバイトデータマルチプレクサ362、363、364、および365と、ストアバッファエントリ339のタグに対応するストアバッファエントリタグマルチプレクサ370と、ストアバッファエントリマルチプレクサ339のアドレスに対応するストアバッファエントリアドレスマルチプレクサ372とともに、ストアバッファエントリレジスタ360を含む。ストアバッファエントリ回路302はまた、タグ比較回路374とアドレス比較回路376とを含む。ストアバッファエントリレジスタ360は、ストアバッファデータエントリレジスタ380と、ストアバッファアドレスエントリレジスタ382と、ストアバッファタグエントリレジスタ384と、ストアバッファ制御エントリレジスタ386とを含む。

【0079】ストアバッファエントリレジスタ回路360は、ストアバッファエントリデータバイトマルチプレクサ362-365と、タグマルチプレクサ370と、アドレスマルチプレクサ372とからストアバッファエントリ339を並列に受け、かつストアバッファエントリ回路SB1およびSB3に並列にストアバッファエントリ339を与えるレジスタである。さらに、ストアバッファデータエントリレジスタ380は、保留ステーションミキサ回路220のデータポートAおよびデータポートBにデータバイト0-3を与える。これらのデータバイトはロードストア機能ユニット134によるロード転送動作の実行を可能にするために与えられる。

【0080】バイトマルチプレクサ回路362-365は、Aマージ回路306、Bマージ回路308、および4つの結果バスならびにストアバッファエントリ回路SB3、SB0およびSB1からそれぞれバイトを受取る。バイトマルチプレクサ回路362-365はストアバッファ制御信号によって制御され、これらのストアバッファ制御信号は、各ストアバッファエントリ毎の線形アドレス、および保留ステーション中のエントリからの線形アドレスの一致に基づいてロードストアコントロール182によって与えられる。結果バスはストアバッファ制御信号によって制御され、これらのストアバッファ制御信号は特定のバイトについてタグ有効ビットが存在するかどうかに基づいてロードストアコントロール182によって与えられる。もしタグ有効ビットが特定のバイトについてセットされると、その特定のバイトは結果バスをモニタし、かつタグに一致する値を有する結果バスであればどれでもマルチプレクスする。

【0081】たとえば、バイトマルチプレクサ回路36

2は、Aマージ信号と、Bマージ信号と4つの結果信号と、ストアバッファエントリSB3、SB0およびSB1との各々からバイト0データを受取る。ストアバッファ制御信号に基づいて、バイトマルチプレクサ回路362はこれらのデータバイトの1つをストアバッファレジスタ回路360中に保持されるSB2ストアバッファエントリとして与える。

【0082】ストアバッファデータレジスタ380にストアされる各バイトはメモリにストアされているものを直接反映するため、データバイトをメモリにストアされているものに対応させるようにバイトステアリングが設けられる。バイトステアリングは、バイトマルチプレクサ0 362およびバイトマルチプレクサ1 363に4つの結果バスバイト0および4つの結果バスバイト1からの入力を並列に与え、バイトマルチプレクサ2 364に4つの結果バスバイト0、4つの結果バスバイト1および4つの結果バスバイト2からの入力を並列に与え、かつバイトマルチプレクサ3 365に4つの結果バスバイト0、4つの結果バスバイト1および4つの結果バスバイト3からの入力を並列に与えることによって設けられる。結果信号のLバイトおよびMバイトはストアバッファ中のどのバイト位置にも対応し得るため、マルチプレクサ2および3 364、365は結果バスバイト0および1を受取る。しかしながら、結果バイト2はデータバイト2にのみ対応でき、かつ結果バイト3はデータバイト3にのみ対応できる。

【0083】アドレスマルチプレクサ372は、保留ステーション124からADDR A信号とADDR B信号とを受取り、これらのアドレスのうちの1つを線形アドレス342としてストアバッファアドレスレジスタ382に与える。ストアバッファアドレスレジスタ382は、ストアバッファエントリ339のアドレス部分342をアドレス比較回路372に与え、アドレス比較回路372はまた保留ステーション124からADDR A信号とADDR B信号とを受取る。アドレス比較回路372はADDR A信号およびADDR B信号を各クロックサイクル毎に線形アドレス342と比較する。ADDR AまたはADDR Bと線形アドレス342との間が一致すれば、ロードストアコントロール182は保留ステーション124にストアバッファデータレジスタ380からデータを読み出させ、これはデータキャッシュ150の対応するポートではなくてアドレス比較一致に対応するポートを介して行なわれる。

【0084】タグマルチプレクサ370はストアバッファエントリSB0、SB1およびSB3からタグを受取る。タグマルチプレクサ370はまた保留ステーションエントリのAタグおよびBタグからタグを受取る。タグバイトはタグレジスタ384に保持され、転送されるが、タグレジスタ384は結果バスからタグ入力を受取らない。結果バスからのタグはタグ制御回路374によ

ってモニタされる。もしタグレジスタ384によって保持されるタグが結果バスの1つからのタグと一致すれば、タグ制御回路374は、タグの一致を与える結果バスがデータに対応するストアバッファデータレジスタへ与えるようにバイトマルチプレクサ362-365を制御する。

【0085】ストアバッファエントリ339の制御部分344はロードストアコントローラ182によってストアバッファ制御レジスタ386に与えられる。

【0086】ストアバッファエントリ回路SB0、SB1およびSB3の唯一の相違点は、他のストアバッファエントリから入力信号が与えられることである。より特定的には、ストアバッファエントリSB0はストアバッファエントリSB1からの出力だけを受取る。ストアバッファエントリSB1はストアバッファSB0およびSB2からの出力エントリを受取る。ストアバッファエントリSB3はストアバッファSB0、SB1およびSB2からの出力エントリを受取る。

【0087】図6-図8を参照して、ストアバッファ180は係属中のストア動作を一時的にストアする。ストアバイトタグを用いることによって、これらの係属中のストア動作は必ずしも完全なストア動作を有する必要がない。さらに、ストアバッファエントリフィードバックとともにストアバイトタグを用いることによって、ストアバッファ180はストア転送動作を実行する。さらに、ロード動作はデータキャッシュ150にまだストアされていないストア動作に依存するかもしれないため、ストアバッファ180はロード転送動作を実行可能である。

【0088】たとえば、係属中の32ビット更新に伴うレジスタの倍長語のストアのために、各タグ有効ビットによって示されるように保留ステーションエントリ中のバイトタグ0-3が有効である。機能ユニットがストア動作用の値を生成しようとしているがまだ生成していない場合に、更新が係属中であると呼ぶ。もしキャッシュアクセスによってキャッシュヒットが与えられ、ストア動作は保留ステーションエントリRS0からストアバッファ回路180へに移る。保留ステーションエントリのオネベラント上位バイトタグATAGUは、ストアバッファエントリ中のバイト3およびバイト2タグとして複製される。ATAGLおよびATAGM保留ステーションバイトタグは、それぞれストアバッファバイト0タグおよびバイト1タグとして与えられる。(倍長語の書込の場合、これらのタグのすべては実際には同一である。)バイト制御ビットB0およびB1のどちらもセットされない。機能ユニットによって結果が利用可能となると、ストアバッファ180はタグ比較回路374を用いて各バイトタグを結果バス上に現われるタグと比較し、かつマルチプレクサ362-365を用いてタグが一致するときはいずれも結果バスの各バイトからのデー

タをゲート入力する。倍長語のストアの際には各バイトは同時に一致する。

【0089】係属中の倍長語およびそれに続く同じ倍長語の1バイトに対する係属中のバイト更新に伴うレジスタへの倍長語の記憶のためには、少なくとも2つのタグが最終倍長語に現われる。同一のタグがバイト0、2および3について用いられ、かつバイト1については異なるタグが用いられる。この異なるタグは第2のバイトストアが発生したことを表わす。より特定的には、第1の倍長語は4つの有効タグとともにストアバッファエントリSB0にストアされ、かつバイトストアは新しいタグがバイト1に位置した状態で上位ストアバッファエントリSB1中にストアされ、一方、バイト0、2および3からのタグがSB0から転送される。したがって、結果バス上にバイト1結果を与え、バイト制御ビットを用いてバイト1結果をステアリングなしに倍長語ストアバッファエントリ中へ書込むストア転送が達成される。

【0090】メモリ中のバイト2およびバイト3への係属中の更新に伴うワードレジスタへのワードストアのためには、B1ビットがバイト3中でセットされB0ビットがバイト2中でセットされた状態でバイト0およびバイト1のためのタグがバイト2およびバイト3中へ書込まれる。このタグが結果バス上に駆動されると、これらのバイトはそれぞれ、このタグに対応する結果バスのバイト0およびバイト1からストアバッファデータレジスタ380へと同時に転送される。この例はまた、ストアバッファエントリにストアされる1ワードに対して2つの係属中のバイト更新が存在する場合に当てはまる。ストアバッファエントリ中の2つのバイトはおそらく異なる時間に異なる結果バスから転送される。

【0091】バイトストアについては、ソースバイトが上位バイトであるか下位バイトであるかに依存してB1ビットまたはB0ビットがセットされるタグと1つのバイトとが取換えられる。このタグが一致すると、結果バスの示されたバイトからのデータをゲートする。このことは係属中の一語または倍長語更新を有するレジスタのバイトストアの場合にも当てはまる。この場合、バス全体が有効データを含み得るとしても、バイトは結果バスの対応する位置にあることが予想される。

【0092】ストア動作を実行している場合、ストアの読出局面でデータキャッシュ150ではなく下位ストアバッファエントリから転送されたデータを受取ることもある。その結果、ストアバッファ180は既にタグを持っているデータ語の中へタグを挿入する。これはたとえば1つ以上のバイトが短い時間間隔の間に同一の倍長語中へ書込まれる場合に発生する。したがって、ストアバッファエントリにストアされる情報は、その各々が異なる結果を表わす1つ以上のタグを持つことが可能である。動作中には、各タグは結果バスとの比較を行ない、適切な時間に適切なバイトをゲート入力する。未整列の

ストアはストアバッファ180の中へタグを書込むことはできないので、おかしな転送は発生しない。

【0093】ロード動作を実行する場合、ストアバッファ180のアドレス比較回路376は、RS0およびRS1加算器によって与えられる線形アドレスをストアバッファエントリの線形アドレスと比較する。アドレス比較回路376が与えるヒット信号が示すように、ロードアドレスとストアバッファエントリの1つにストアされたアドレスとが一致すると、ロードストアコントロール182はロードがストアに依存しているかと判断する。もしロードがストアに依存していれば、線形アドレスの一致を与えたストアバッファエントリからのデータが、アドレスの一致が与えられたいずれかのポートを介して与えられる。この動作はロード転送動作と呼ばれる。

【0094】図9を参照して、データキャッシュ150は線形にアドレス指定されたキャッシュである。引用により援用される「線形アドレス指定可能なマイクロプロセッサキャッシュ (Linearly Addressable Microprocessor Cache)」と題された同時出願の米国特許出願連続番号第146,381号は、データキャッシュ150の線形アドレス指定についての構造および動作をより詳細に説明している。

【0095】データキャッシュ150のエントリ400が示される。データキャッシュ150の各エントリごとに、キャッシュエントリに対応する各線形アドレスのうちの中位ビットはキャッシュインデックスを与え、このキャッシュインデックスは線形タグアドレスをアドレス指定し、かつ各線形タグアドレスからエントリを取出すために用いられる。各線形アドレスの上位ビットは、アドレスタグアドレス310から取出されたエントリ内にストアされる線形データタグと比較される。各線形アドレスの最下位ビットは取出されたエントリへのオフセットを与えて、線形アドレスによってアドレス指定された実際のバイトを見つける。データキャッシュ150は常に32ビットワードの形であるため、これらの最下位ビットはデータキャッシュ150にアクセスする場合には使用されない。

【0096】データキャッシュ150のデータキャッシュエントリ400は線形アドレスタグエントリ402とデータエントリ404とを含む。データエントリ404は16バイト (BYTE0-BYTE15) ブロックのデータを含む。データ線形アドレスタグエントリ402は、データ線形タグ値 (DTAG) と、線形タグ有効ビット (TV) と、有効物理変換ビット (P) とを含む。線形アドレスの上位21ビットに対応するデータ線形タグ値は、対応するストアアドレスエントリにストアされるブロックの線形ブロックフレームアドレスを示す。線形タグ有効ビットは線形タグが有効であるかどうかを示す。有効物理変換ビットはエントリが物理タグヒットをうまく与えられるかどうかを示し、これについては以

下に説明する。

【0097】図10を参照すると、線形にアドレス指定可能なデータキャッシュ150のデータキャッシュ線形タグ回路202とデータキャッシュストアアドレス200とが示される。データキャッシュ150は4つの2Kバイトカラムに、つまりカラム0、カラム1、カラム2、およびカラム3に配列される。データ線形タグ回路202は2つの線形アドレスADDR AとADDR Bとを同時に受取り、データストアアドレス200は2つのデータ信号DATA AとDATA Bとを同時に与える。つまりデータキャッシュ150は二重にアクセスされるデータキャッシュとして作用する。

【0098】データストアアドレス200は4つの別個のデータストアアドレス、つまりカラム0ストアアドレス430、カラム1ストアアドレス431、カラム2ストアアドレス432、およびカラム3ストアアドレス433とともにマルチプレクサ (MUX) 回路440を含む。マルチプレクサ440はデータ線形タグ回路202から制御信号を受取り、これらの制御信号は各線形タグアドレス中にストアされる線形タグ値と一致するかどうかを示す。マルチプレクサ440はストアアドレス430-433からデータを受取り、かつこのデータをロードストア機能ユニット134に与える。

【0099】線形タグ回路202はカラム0-3に対応する線形タグアドレス450-453を含む。各線形タグアドレスは対応する比較回路454-457に結合される。したがって、データキャッシュ150の各カラムはストアアドレスと、線形タグアドレスと、比較回路とを含む。ストアアドレス430-433、アドレスタグアドレス450-453、および比較回路454-457はすべてロードストアセクション134から線形アドレスADDR A、ADDR Bを受取る。

【0100】IADバス102はストアアドレスマルチプレクサ461を介して各ストアアドレス430-433に結合されていずれにもストアアドレスを与える。IADバス102はまた、各ストアアドレス430-433に結合されるストアレジスタ460に結合される。IADバス102によって与えられるストアアドレスは、特定のカラムを指して特定のバンクを選択するために与えられ、特定のカラムはカラム選択ビットによって選択され、これらのカラム選択ビットはストアを行なっているときにストアバッファ180によって与えられるか、または再ロードを行なっているときに物理タグ回路162によって与えられる。ストアのためには、ただ1のバンクがアクセスされる。バンク選択ビット、つまりIADバス102によって与えられるアドレスのビット2およびビット3はバンクにアクセスするために用いられる。再ロードのためには4つのバンクすべてが並列にアクセスされる。

【0101】IADバス102はデータをデータキャ

シュ150のストアアレイ430-433に書込むために、ストア動作および再ロード動作の両方の動作の間に使用される。ストア動作を実行するときには、データは32ビット倍長語の形でストアレジスタ460を介してストアアレイ430-433に書込まれる。ストアバッファの書込のためには、ADDR Bに与えられるIA Dバスアドレスがデータキャッシュ150に入力される。ADDR BおよびIADアドレスはアドレスマルチプレクサ461によってマルチプレクスされる。

【0102】再ロード動作を行なうときには、データは128ビットラインでストアアレイ430-433に書込まれる。ストアレジスタ460は2回の64ビットアクセスでIADバス102から128ビットのデータを集め、この128ビットが集められた後、ストアレジスタ460はこのデータをストアアレイ430-433に書込む。再ロードのためには、64ビットが各位相で書込まれるため、ストアレジスタ460はデータを受取るためにIADバス102のアドレスラインをマルチプレクスする。アドレスマルチプレクサ461は、ロウを指し示すためにIADアドレスをADDR Bアドレス経路上にマルチプレクスする。データキャッシュストアマルチプレクサ460は、ストア動作またはロード動作のどちらが行なわれているのかに基づいてデータキャッシュコントローラによって制御される。再ロード動作のためには、ロードストアポート134はデータキャッシュ150のポートAを介して再ロードアドレスを書込み、このためデータキャッシュ150は再ロードアドレスのためにADDR Aを用いる。

【0103】図1および図12を参照すると、データキャッシュ150の各ストアアレイは、デュアルポート動作に関連したオーバーヘッドを必要とせずに、1クロックサイクル中に複数のアクセスが可能となるようにバンク構成とされる。より特定的には、各ストアアレイは4つのバンク470-473を含む。これらのバンクの各々は32ビット倍長語データをストアし、各バンクはそれぞれのバンクアドレスマルチプレクサ474-477を含む。4つのバンクの組合わせによりデータキャッシュ150の1ラインへのアクセスが与えられる。

【0104】各バンク470-473はそれぞれADDR AまたはADDR Bのいずれかによってアドレス指定され、これらのアドレスは各バンクアドレスマルチプレクサ474-477によって与えられる。バンクアドレスマルチプレクサ474-477はADDR AおよびADDR Bのバンク選択ビットによって制御される。各バンクは個々にアドレス指定されるため、1つ以上のバンクを同時にアクセスし得る。

【0105】たとえば、図11に示されるように、ADDR Aがバンクの1ラインをアドレス指定し、ADDR Bがバンク3の同じラインをアドレス指定する場合、マルチプレクサ474はADDR Aをバンク0に

与え、かつマルチプレクサ476はADDR Bをバンク2に与える。ADDR Aによってアドレス指定されたデータ語は、DATA Aデータ経路を介してDATA A Aとしてロード/ストア機能ユニット134に与えられ、ADDR Bによってアドレス指定されたデータ語は、DATA Bデータ経路を介してDATA Bとしてロード/ストア機能ユニット134に与えられる。

【0106】図12からわかるように、ADDR AおよびADDR Bの両方がバンク0の同じラインをアドレス指定する場合、このラインとバンクとだけがアクセスされ、この位置のデータはDATA Aデータ経路およびDATA Bデータ経路をそれぞれ介してDATA AおよびDATA Bの両方としてロード/ストア機能ユニット134に与えられる。

【0107】2つのアクセスがバンクは同じだが異なるラインへのアクセスである場合、データキャッシュコントローラ190によって1サイクルの間ポートBアクセスが停止される。局所性が強い命令キャッシュアクセスと比べるとデータキャッシュアクセスは一般にランダムであるため、同じバンクの異なるラインへのポートA、ポートBのアクセスが発生する頻度は相対的に低い。

【0108】データキャッシュ150へのストアアクセスはIADバス102を介して行なわれる。ストアの間、マルチプレクサ474-478は、バンク470-473のうちのどれに32ビットストア倍長語が書込まれるかを制御するためにストアアクセスを使用する。再ロードの間、バンク470-473は再ロードデータがストアレジスタ460中に集められた後に1つの128ビットラインに書込まれる。

【0109】図2、および図9-図11を参照して、データキャッシュ150の一般的な動作について議論する。ロード/ストア機能ユニット134がキャッシュ150にストアされていないデータ値をリクエストすると、キャッシュミスが発生する。キャッシュミスを検出すると、リクエストされた値がデータキャッシュ150のエントリに書込まれる。より特定的には、ロードストアセクション134はその値についての論理アドレスを線形アドレスに変換する。この線形アドレスはメモリ管理ユニット164に与えられる。TLB比較回路はこの値の線形アドレスをメモリ管理ユニットのTLBアレイの線形タグ部分と照会して、TLBヒットが存在するかどうかを判断する。

【0110】ロードストア機能ユニット134がTLBヒットがあると判断すると、ロードストア機能ユニット134はデータを検査してデータがキャッシュ可能かどうかを判断する。もしデータがキャッシュ可能であり、かつTLBヒットがあれば、対応する物理アドレスの物理タグが物理タグ回路162の対応するエントリの中へ書込まれる。データがストアされたアレイカムに対応するデータ線形タグアレイ450-453にはTLB A

レイからの線形タグが書込まれる。

【0111】TLBヒットがなければ、TLBアレイはTLBヒットが生じるように、メモリ管理ユニット164によって要求された値のアドレスを含むように更新される。その後、物理タグが物理タグ回路162に書込まれ、線形タグが適切な線形タグアレイ450-453に書込まれる。

【0112】その後、ロード/ストア機能ユニット134が外部メモリに対してプリフェッチ要求を行ない、外部メモリ中の線形アドレスに対応する物理アドレスにストアされた値が外部メモリから取出される。この値はストアアレイ200のバンク、ラインおよびカラムにストアされており、ストアアレイ200は線形タグアレイにストアされる値の線形タグのライン位置およびカラム位置に対応する。線形タグアレイ310中の対応する線形タグ有効ビットおよび有効物理変換ビットは、線形タグに対応するエントリが有効であり、線形タグが有効であり、かつエントリが物理変換をうまう行なうことを示すようにセットされる。

【0113】ロード/ストア機能ユニット134がこの値についての線形アドレスを再び要求すると、ロードストアセクション134は論理アドレスを線形アドレスに変換し、この線形アドレスは要求されたアドレスと線形アドレスタグアレイ310中の線形タグとの一致を与える。有効ビットがセットされており有効物理変換ビットがセットされているため、線形アドレスヒットが発生し、かつデータストアアレイ304の対応するラインにストアされるエントリがロード/ストア機能ユニット134に与えられる。ロードストアセクション134によるアクセスの間、有効物理変換ビットがセットされてエントリが有効物理変換を有することを示しているため、物理アドレスタグ回路162またはTLB回路164のどちらへもアクセスの必要がない。

【0114】図1-図10および図13を参照して、ロード/ストア機能ユニット134がポートAを介してロード動作を実行しており、かつロードされるべきデータ値がデータキャッシュ150中で利用可能な場合、データキャッシュヒットが発生する。より特定的には、周期1のΦ1の間に加算器240またはRS0加算器216によって計算されたキャッシュインデックスが発生する。このキャッシュインデックスは線形アドレスの最下位11ビットであり、線形アドレス計算の一部として計算される。このキャッシュインデックス線形アドレスはデータキャッシュストアアレイ200の適切なラインおよびバンクにアクセスするために用いられる。適切なラインおよびバンクにアクセスするときには、加算器242によって計算された線形アドレスが線形タグと比較することによってストアアレイ200の適切なカラムにアクセスするために使用される。その後、データ値はDATA Aデータ経路を介して保留ステーション回路124

のドライバ回路220に戻される。このデータ値はドライバ回路220によってフォーマット化されて結果バス0に与えられる。周期1のΦ2の間、リミットチェック回路252は当該技術分野で周知のように線形アドレスに対してセグメントリミットチェックおよび保護チェックを行なう。周期2のΦ1の間、データ値および対応する先行タグがポートAのために結果バス0上に駆動される。

【0115】ポートAを介してロード動作が実行されている間、対応するロード動作がポートBを介して実行され得る。この対応するロード動作はデータキャッシュアクセスのアドレス発生を行なうために、対応する加算器とともに保留ステーションRS1を用いる。保留ステーションRS1中のエントリについてのデータ値および対応する先行タグは結果バス1上に駆動される。

【0116】図1-図10および図14を参照して、ロード/ストア機能ユニット134によってポートAを介してストア動作が実行されており、かつストアされるべきデータ値がデータキャッシュ150中に既にストアされている場合、データキャッシュヒットが発生する。ストアは読出修正書込動作として実行されるため、ストア動作の第1の部分はロード動作と同様である。データ値がロードされた後、ロードされた値はロードされたデータ値を修正するためにストアバッファ回路180に書込まれる。

【0117】より特定的には、周期1のΦ1の間、加算器240またはRS0加算器216による計算によってキャッシュインデックスが発生される。このキャッシュインデックスは線形アドレスの最下位11ビットであり、線形アドレス計算の一部として計算される。このキャッシュインデックス線形アドレスはデータキャッシュストアアレイ200の適切なラインおよびバンクにアクセスするために用いられる。適切なラインおよびバンクがアクセスされるときには、線形タグと比較することによってストアアレイ200の適切なカラムにアクセスするために、加算器242によって計算された線形アドレスが用いられる。その後、データ値はDATA Aデータ経路を介して保留ステーション回路124のドライバ回路220に戻される。このデータ値はドライバ回路220によってフォーマット化されて結果バス0に与えられる。周期1のΦ2の間、リミットチェック回路252は当該技術分野で周知のように線形アドレスに対してセグメントリミットチェックおよび保護チェックを行なう。周期2のΦ1の間、データ値および対応する先行タグがポートAのために結果バス0上に駆動され、かつまたストアバッファ回路180の次に利用可能なエントリにストアされる。この値はストア動作がリオーダーバッファ114からリタイアするまでストアバッファ回路180に保持され、ストア動作のリタイアはどの命令も関係していない場合に発生する。その後、リオーダーバッファ

114はロードストアリタイア信号を用いてロード/ストアコントロール180に対してストア命令をリタイアできる。つまりストアを実行できるということを示す。ストアはデータ値の状態を実際に修正するため、ストアは仮想に基づいては実行されず、リオーダーバッファ114がストアの実行を許可する前にストアが実際に次の命令であるということがはっきりするまで待機しなくてはならない。

【0118】リオーダーバッファ114が命令を実行してもよいということを示した後、命令の解除に続いて周期のΦ1の間、データ値および対応する線形アドレスがIADバス102に対して駆動される。この周期のΦ2の間、データ値はデータキャッシュストアアレイ200の適切なラインおよびバンクに書込まれる。さらに、もし物理タグ回路162がこの値を外部にもまた書込むべきであるとせば、データ値は線形アドレスに対応する物理アドレス位置において外部メモリに書込まれる。IADバス102から線形アドレスをまた受取るメモリ管理ユニット164によって物理アドレス変換が行なわれる。

【0119】図1-図10および図15を参照して、ロード/ストア機能ユニット134が仮想に基づくロード動作を実行しており、かつロードされるべきデータ値がデータキャッシュ150中で入手できない場合、仮想に基づくデータキャッシュミスが発生する。ロード動作の第1の周期はキャッシュヒットが発生した場合と同じである。

【0120】キャッシュ150がアクセスされ、キャッシュミスが生じた場合、周期2の間にメモリ管理ユニット164中でTLBがアクセスされ、かつデータ値の物理アドレスを決定するために物理タグ回路162中で物理タグがアクセスされる。その後、この物理アドレスは保護チェックのどれにも違反しないことを確認するためにメモリ管理ユニット164内でチェックされる。次の周期の間、ポートBアクセスがキャッシュアレイ200の同じバンクへのアクセスではない場合、ポートBは他のキャッシュアクセスを開始する。さらに、このサイクルのΦ2の間、タグバスからのラインのタグ有効ビットを用いてキャッシュアレイ200が更新される。次の周期の間、データ値、行先タグおよびステータスが次に利用可能な結果バス上に駆動され、かつキャッシュヒットを想定した通常の動作が始まる。

【0121】図1-図10および図16を参照して、キャッシュ再ロードの間、再ロード動作の第1の周期はキャッシュヒットが生じた場合と同一である。しかしながら、キャッシュコントロール190がキャッシュミスが生じたと判断した後は、ロード/ストア機能ユニット134は、外部メモリを再ロードキャッシュ150にアクセスする前にストアバッファ回路180が空になるのを待つ。数クロック周期待機した後、物理タグ回路16

2は、データの128ビットすべてがストアレジスタ460に書込まれたことをキャッシュ150に対して示すデータ使用可能信号(L22LS)を与える。データが使用可能となりデータキャッシュアレイ200に書込まれると、保留ステーション回路124のドライバ回路220はデータ、行先タグおよびステータス情報を結果バス0上に駆動する。

【0122】図17を参照して、誤整列したアクセスについては、続く周期の間に2つのアクセスが存在する。2つのアクセスの各々はキャッシュヒットアクセスと同じである。各アクセスから戻ったデータはドライバ回路220によって集積される。2つのアクセスが完了し、データが集積された後、ドライバ回路220は上に述べたようにデータをフォーマット化する。その後、保留ステーション回路124のドライバ回路220はデータ、行先タグおよびステータス情報を結果バス0上に駆動する。誤整列したアクセスは保留ステーション0を用いる場合にのみ実行される。したがって、ドライバ回路220のRS0加算器およびポートA部分だけが、誤整列したアクセスの実行に必要な回路を必要とした。

【0123】他の実施例

他の実施例は前掲の特許請求の範囲内である。

【0124】たとえば、ロード/ストア機能ユニット134はロード機能ユニットとストア機能ユニットとの2つの別個の機能ユニットに分割されてもよい。この実施例では、これらの機能ユニットの動作は上述とほぼ同じであるが、各機能ユニットはそれぞれの保留ステーションを含み得る。言い換えれば、ロードセクションはロードに関して議論したように機能するロード保留ステーションを含み、かつストアセクションはストアに関して議論したように機能するストア保留ステーションを含む。

【図面の簡単な説明】

【図1】本発明に従うスーパーバスカラムマイクロプロセッサのブロック図である。

【図2】本発明に従うロード/ストア機能ユニットおよびデータキャッシュのブロック図である。

【図3】図2のロード/ストア機能ユニットの保留ステーション回路のブロック図である。

【図4】図3の保留ステーション回路のエントリの内容のブロック図である。

【図5】図3の保留ステーション回路の加算器回路のブロック図である。

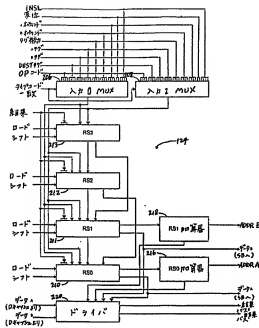
【図6】図2のロード/ストア機能ユニットのストアバッファ回路のブロック図である。

【図7】図6のストアバッファ回路のエントリの内容のブロック図である。

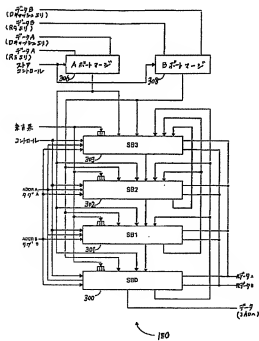
【図8】図6のストアバッファ回路のストアバッファエントリのブロック図である。

【図9】図2のデータキャッシュのエントリのブロック図である。

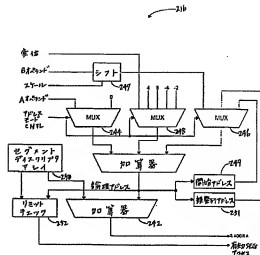
【図3】



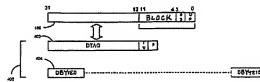
【図6】



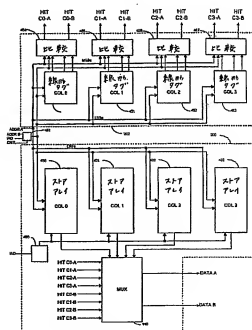
【図5】



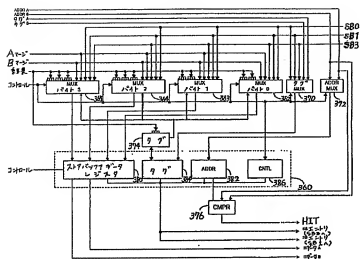
【図9】



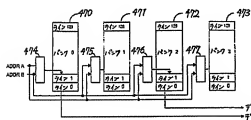
【図10】



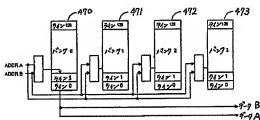
【図8】



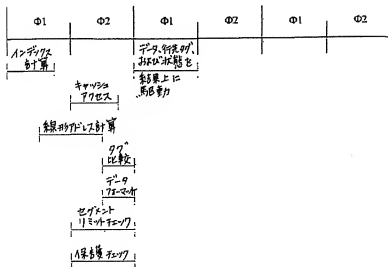
【图 1-1】



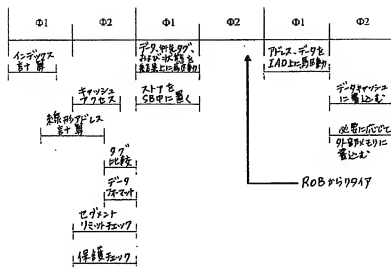
【圖12】



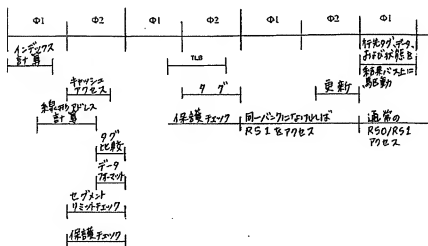
【图13】



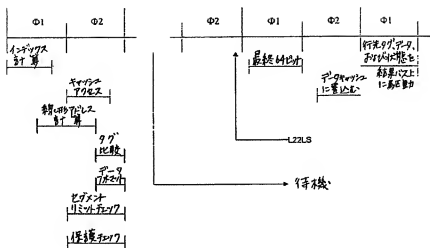
【図14】



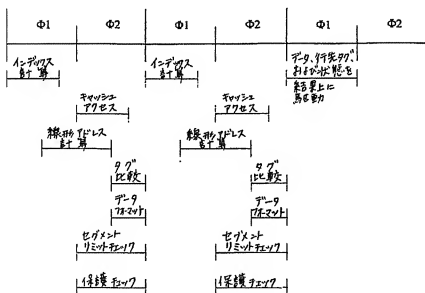
【図15】



【図16】



【図17】



フロントページの続き

(72)発明者 ウィリアム・エム・ジョンソン
アメリカ合衆国、78746 テキサス州、オ
ースティン、クリスティー・ドライブ
102

(72)発明者 デイビッド・ビー・ウィット
アメリカ合衆国、78759 テキサス州、オ
ースティン、バスファインダー・ドライ
ブ、6318

(72)発明者 ミュラリ・チナコンダ
アメリカ合衆国、78746 テキサス州、オ
ースティン、スパイグラス・ドライブ
1781、ナンバー・301